

CONTRACT NAS 9-10441
FINAL REPORT

(NASA-CR-115606) DATA COMPRESSION/ERROR
CORRECTION DIGITAL TEST SYSTEM. APPENDIX 2:
THEORY OF OPERATION Final Report
(Radiation, Inc.) Jan. 1972 163 p CSCL

N72-25231

Unclas
09B G3/08 31301

DATA COMPRESSION/ ERROR CORRECTION DIGITAL TEST SYSTEM

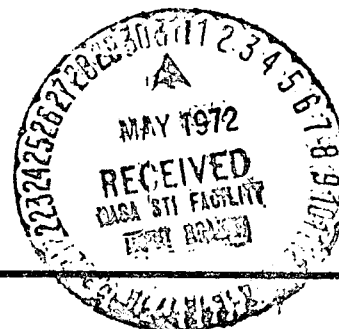
APPENDIX 2 ✓

THEORY OF OPERATION

OFFICE OF PRIME RESPONSIBILITY

JANUARY 1972

~~U.S. GOVERNMENT AGENCIES AND
CONTRACTORS ONLY~~



86185-F



RADIATION
Systems Division
MELBOURNE, FLORIDA 32901

SUBSIDIARY OF HARRIS-INTERTYPE CORPORATION

DATA COMPRESSION/ ERROR CORRECTION DIGITAL TEST SYSTEM

APPENDIX 2

THEORY OF OPERATION

JANUARY 1972

TABLE OF CONTENTS

<u>Paragraph</u>	<u>Title</u>	<u>Page</u>
A2-1.0	INTRODUCTION	2
A2-1.1	System Block Diagram Description	2
A2-2.0	DETAIL TRANSMITTER DESCRIPTION	5
A2-2.1	Filter - FL1	5
A2-2.2	Video Amplitude Shaper - A3 A2 A2	5
A2-2.3	Analog-to-Digital Converter - A1	11
A2-2.4	Sync Separator - A3A2A1	12
A2-2.5	Reducer - A3A1A3	12
A2-2.6	Statistical Encoder - A3A5	19
A2-2.7	Transmit Rate Buffer Subsystem - A3A1A4	26
A2-2.7.1	Rate Buffer	26
A2-2.7.2	Detail Operation	26
A2-2.8	Rate Adaptive Tolerance Generator - A3A1A1	29
A2-2.8.1	Detailed Description	29
A2-2.9	Voice Encoder - A3A3A4	31
A2-2.10	Information Rate Timing - A3A2A6	33
A2-2.11	Sample Rate Clock - A3A2A6(A7)	35
A2-2.12	Time Division Multiplexer - A3A2A1	37
A2-2.12.1	Format	37
A2-2.12.2	Frame Rate Counter	40
A2-2.12.3	Data Source	40
A2-2.12.4	Multiplexing Operation	41
A2-2.13	Convolution Encoder - A3A3A3	41
A2-2.13.1	Detail Description	42
A2-2.14	Noise Generator - A3A3A2	44
A2-2.15	Output Buffer Amplifier - A3A3A1	44
A2-3.0	DETAIL RECEIVER DESCRIPTION	45
A2-3.1	Bit Sync-Signal Conditioner - Preprocessor - A1(A2)	45
A2-3.2	Quadrature Data Combiner - A9A2	49
A2-3.3	Viterbi (Maximum Likelihood) Decoder	50
A2-3.3.1	Review of Decoding Process	50
A2-3.3.2	General Functional Organization of the Decoder	53
A2-3.3.3	Block Diagram Description	56
A2-3.4	Fano Decoder - A8A1A9 (A13-A22)	97
A2-3.4.1	Theory of Operation	97

TABLE OF CONTENTS (Continued)

<u>Paragraph</u>	<u>Title</u>	<u>Page</u>
A2-3.5	Group Synchronizer and Demultiplexer - A8A1A2	134
A2-3.5.1	Group Sync Policy	134
A2-3.5.2	Group Synchronizer Block Diagram	134
A2-3.6	Receiver Rate Buffer (A8A1A1, A8A1A8, and A8A1A11)	136
A2-3.6.1	Receiver Buffer Operation	136
A2-3.6.2.	Buffer Synchronization	139
A2-3.7	TV Data Reconstructor - A8A1A7	140
A2-3.8	Statistical Decoder - 2A8A6A1 & 2	143
A2-3.9	TV Sync Generator - A8A1A10.	146
A2-3.10	Digital-to-Analog Converter - A8A1A5 and A8A1A7.	146
A2-3.11	Video Amplitude Shaper (Expander) - A8A1A6	146
A2-3.12	Voice Demodulator - A8A1A3	154
A2-3.13	Timing Generator - A8A1A4	155
A2-3.14	Bit Comparators - A8A9A1	157

APPENDIX 2
THEORY OF OPERATION

A2-1.0 INTRODUCTION

This volume consists of three major sections: a system block diagram description, a description of the units of the Transmitter, and a description of the units of the Receiver.

A2-1.1 System Block Diagram Description

An overall block diagram of the DC/EC Digital System Test is shown in Drawing 528769. The system is divided into two major units, the Transmitter and the Receiver. In operation, the Transmitter and Receiver are connected only by a real or simulated transmission link. The system inputs consist of: 1) Standard format TV video, 2) Two channels of analog voice and 3) One serial PCM bit stream.

Considering first the TV channel, the analog video is supplied from a TV camera in synchronism with the sync separator and routed to the Input Filter. The Sync Separator generates the sync and blanking pulses for the video and produces digital sync words. The unit also produces control signals to the Data Reducer to prevent unnecessary signal processing and transmission during sync and blanking times. The Amplitude Shaping unit processes the analog video so that it may be efficiently sampled and quantized by the Analog to Digital Converter. The digital words from the converter are supplied to the Data Reducer. This unit processes the data to eliminate samples which are unnecessary to the later reconstruction process. These nonredundant samples are fed to the Rate Buffer to allow conversion from their nonuniform rate of occurrence to a uniform transmission rate. Different picture areas contain different numbers of non-redundant samples, so the input rate to the buffer varies. Since the average input and output rates of the buffer must be equal, it is necessary to provide feedback control of the Data Reducer by the Rate Buffer, to avoid uncontrolled overflow or underflow of the Buffer.

An alternate method of reducing the number of transmitted bits per picture may be selected by utilizing the Statistical Encoder. This technique transmits a variable number of bits for code words representing the difference between adjacent A/D samples. The length of the transmitted code words is preprogrammed as a function of the frequency of occurrence of the input differences. Differences which occur most often are presented by the shortest length code words while differences which occur least often are represented by the longest code words. As the code words are generated they are stored in the Rate Buffer prior to transmission.

The Rate Buffer output forms one input to the Multiplexer, which combines this signal with the other three signals. Two of these other signals are from the two Delta Modulators, which convert the analog voice input signals into efficient digital form. The third input is a serial PCM telemetry signal. A sync pattern is also multiplexed with the data to allow synchronization of the receiving equipment. The digital voice and TV data are synchronized with the PCM to allow simple time division multiplexing of the signals to a single bit stream at the output of the multiplexer. The multiplexer output is fed to the Channel Encoder (Convolution Encoder) which inserts redundancy into the bit stream in a form which allows the receiver to efficiently correct transmission errors due to noise. The final bit stream is fed to the Output Buffer Amplifier, which provides flexible gain and offset for interface with a variety of modulators or simulators.



REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
	CHK BY	CHK BY	ECO NO	

1A3A5
124898, 124899
531022, 531023

1A3A3A2
123955

1A3A3A3
123966
528734

CONVOLUTION
ENCODER

NOISE
GENERATOR

OUTPUT
BUFFER
AMP

GENERAL
PURPOSE
OUTPUT TO
MODULATOR

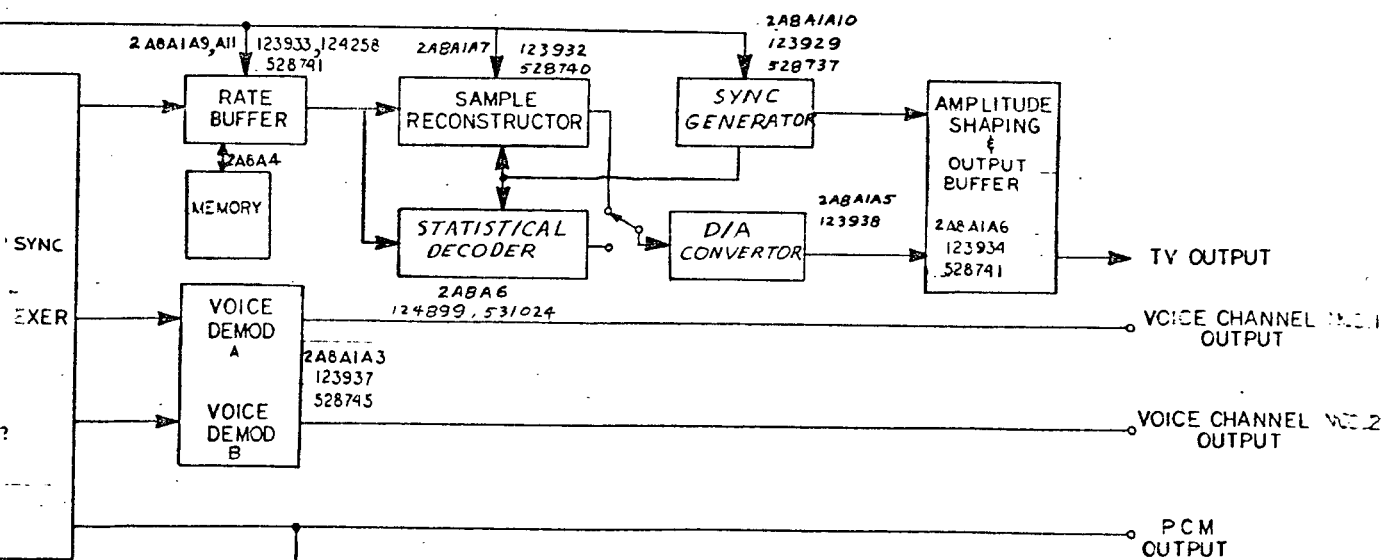
1A3A3A1
123924
528732

LOGIC
OUTPUT

- BI-PHASE
- QUADRI-PHASE

Reproduced from
best available copy.

TRANSMITTER 123914
RECEIVER 123915



ENGRG DEV

QUANTITY REQUIRED		ITEM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	CODE IDENT
LIST OF MATERIALS OR PARTS LIST					
RADIATION INCORPORATED SUBSIDIARY OF HART-S-INTERTYPE CORPORATION, MELBOURNE, FLORIDA					
TITLE					
BLOCK DIAGRAM					
SIZE	CODE IDENT NO	5287-9		REV	
D	91417				
SCALE NONE					
1 SHEET					

PART TO BE FREE OF BURRS AND SHARP EDGES
COMMERCIAL TOLERANCES APPLY TO STOCK SIZES
MANUFACTURING TOLERANCES PER 900002
WORKMANSHIP PER 900060
SCREW THREADS PER MIL-STD-9
ABBREVIATIONS PER MIL-STD-12
LOGIC SYMBOLS PER MIL-STD-806
DIMENSIONING AND TOLERANCING PER USAS
Y14.5-66
ELECTRICAL AND ELECTRONIC DIAGRAM PER USAS
Y14.13-66
GRAPHIC SYMBOLS FOR ELECTRICAL AND
ELECTRONIC DIAGRAM PER USAS Y32.2-67
ELECTRICAL AND ELECTRONIC REFERENCE
DESIGNATIONS PER USAS Y32.16-65
WELDING SYMBOLS PER AWS A2.8-58

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
AND INCLUDE APPLIED FINISH
TOLERANCES

2 PLACE DIM 3 PLACE DIM ANGLES

CONTRACT NO.
ON 27 JULY 70
BY J. J. Russell
ENGINEER
PROJECT ENGINEER
J. J. Russell 12-14-71

APPROVAL
APPROVAL

DASH NO	NEXT ASSY	USED ON

This amplifier also contains provision for mixing noise with the signal to allow link simulation at baseband frequencies. A wideband noise generator with adjustable level is provided. In addition to data, a clock at the transmitted bit rate is provided for use in external equipment.

Directing attention to the receiver section, the input signal is applied to the Bit Synchronizer, which synchronizes a bit rate clock to the input and filters the data with a matched filter. The output of the bit detector filter is sampled (by the derived bit rate clock) and quantized to three bit resolution. In the case of Quadriphase signals a second Bit Synchronizer is provided. The output of the two Bit Synchronizers are combined in the Quadriphase Multiplexer. The combined signal, and clock, are fed to the Decoders which operate on the signal to correct transmission errors and reconvert the signal to the form existing at the Encoder input. The Decoders contain circuitry to establish decoding sync, and to resolve demodulator polarity ambiguities. Two Convolution Decoders are provided. One, a parallel processing device implements the Viterbi algorithm and the other, a sequential device, implements the Fano algorithm. The Decoder output is applied to the Group Synchronizer, which locates the sync pattern inserted at the transmitter and synchronizes the demultiplexer.

The Demultiplexer now separates the signal into four individual channels (and deletes the sync pattern). The PCM is outputted directly, while the two voice channels are reconstructed to analog form in the Voice Demodulators and outputted.

The TV channel words are fed at a uniform rate to the Rate Buffer, and are outputted from the buffer as required by the Reconstructor or the Statistical Decoder. TV sync patterns are recognized and used to synchronize the TV Sync generator, which regenerates the standard TV sync signals. The TV video and sync are converted to analog form in the Digital to Analog Converter. The signal is then shaped in the Amplitude Shaping unit to remove the effect of the shaping performed at the transmitter. The final output is standard TV waveform similar to that at the input of the transmitter.

Two bit comparators are provided to monitor the bit error rate at both the input and output of the Convolution Decoders. These units compare the delayed transmitted data or its equivalent to the corresponding receiver data, and generate error pulses. A Counter counts these errors over an adjustable time period and decimally displays the bit error count.

A2-2.0 DETAIL TRANSMITTER DESCRIPTION

The following sections describe the individual units of the Transmitter in more detail.

A2-2.1 Filter - FL1

The filter is a bandpass filter, manufactured by Krohn-Hite Corporation, Model #3103R. A list of the primary specifications are provided below:

1. Frequency Range: Low-cutoff frequency independently adjustable from 10 Hz to 1 MHz in five bands. High-cutoff frequency independently adjustable from 30 Hz to 3 MHz in five bands.
2. Frequency Dials: Separate low-cutoff and high-cutoff dials.
3. Bandwidth: Continuously variable within the cutoff frequency limits of 10 Hz to 3 MHz.
4. Attenuation Slope: Nominal 24 dB per octave.
5. Frequency Response: Standard response is a 4th order Butterworth, maximally flat. A RESPONSE switch on rear of chassis converts to simple RC response optimum for transient-free performance.
6. Maximum Input Amplitude: 3 volts rms, decreasing to 2.5 volts at 3 MHz.
7. Maximum Output Amplitude: See 6.
8. Maximum Output Current: 10 ma rms.
9. Connectors: Front panel and rear of chassis, one BNC connector for INPUT and for OUTPUT.

A2-2.2 Video Amplitude Shaper - A3 A2 A2

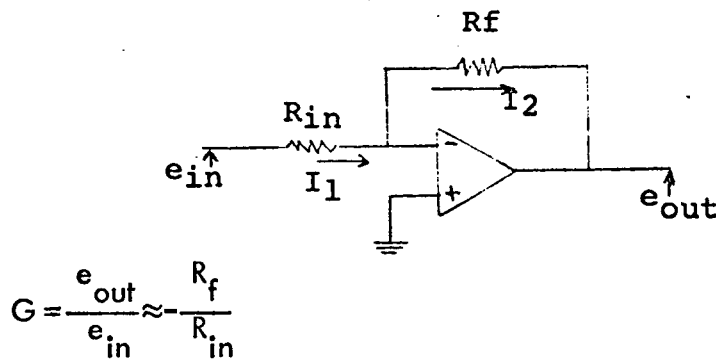
The Transmitter Video Amplitude Shaping Network (Drawing 528730) accepts the composite video input from the system input filter. It amplifies this signal linearly or nonlinearly and provides an output amplitude of 0 to +5 volts to drive the A/D converter. The transfer function of this amplifier is piece wise linear and has up to 3 line segments. The slopes of the line segments are individually adjustable by means of a plug-in resistor module located on the video amplitude shaping board.

The Video Processing Channel consists of amplifiers AR1, AR2, AR3, and AR4 and their associated circuitry. The DC restorer is composed of comparator U1, transistors Q2 and Q3, transformer T1, diodes CR7, CR8, CR10 and CR9 and summing amplifier AR5. Transistor Q1, amplifier AR6 and reference diode CR5 form the necessary circuit to establish a precision voltage to reference the break point of the linear segments. Variable resistor R20 establishes the clamping level of the DC restorer, resistor R27 establishes the bias current for diodes CR2 and CR3 and variable resistors R11 and R14 establish the break points.

Amplifier AR1 is used to drive AR2 and the DC restorer circuit. The input video from AR1 is amplified by AR2 (X5) and shaped by amplifier AR3. AR4 is used to drive the 50 ohm input of the A/D converter.

Amplifier AR3 and its associated circuitry perform the shaping function of the input video. The two break points are determined by the resistive combinations of R13, R14, R35, and R12, R11, R34. The circuit is designed to operate on a positive going signal with diode CR1 removing all negative going signals. Thus, using the level adjustment (R20) the incoming signal can be clamped such that only the sync pulse is removed. A brief review of operational amplifier fundamentals is presented to help clarify the discussion of the shaping network.

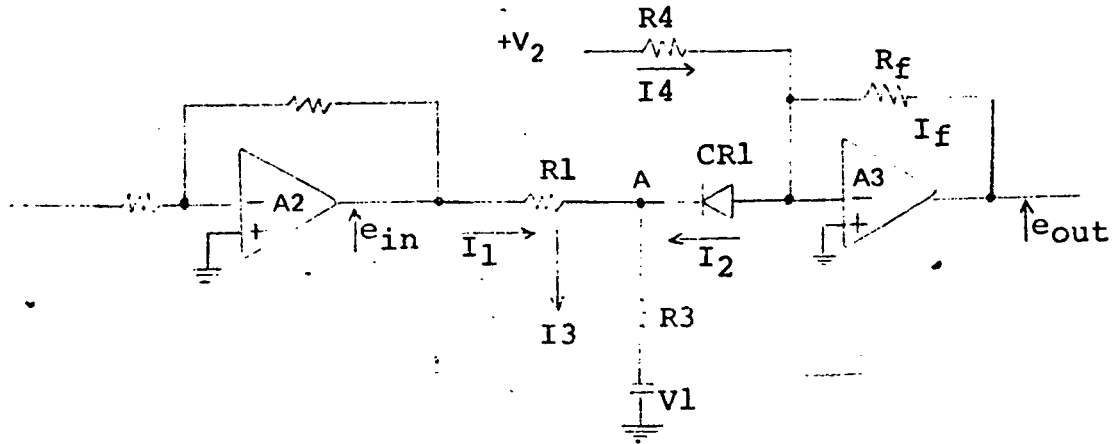
The gain of an operational amplifier is expressed as:



The gain equation is an approximation, but it is reasonably accurate for low frequencies, and provided the amplifier has a large open loop gain. Other approximations to be used in the following discussion are:

$$I_1 = I_2$$

and the summing point for I_1 and I_2 is ground. Therefore, let's consider the following circuit:



For this discussion the break occurs on a positive going signal, as does the actual shaper Figure 3.3-2. The diode CR1 is biased on for 0 volts in. The bias current is determined by V_2 and R_4 . Resistor R_3 and V_1 are adjusted to sink I_2 and I_1 plus establish the break point. Since there is a drop across a forward biased diode, amplifier A2 will have to supply enough current for $e_{in} = 0$, such that $I_1 R_1$ is equal to the diode drop. Therefore, for $e_{in} = e_{out} = 0$ volts. The following conditions exist:

$$I_f = 0$$

$$I_4 = I_2 = \frac{V_2}{R_4}$$

$$I_3 = I_1 + I_2$$

$$I_1 \approx \frac{.6}{R_1}$$

As e_{in} is increased in the positive direction I_2 decreases and $I_f = I_4 = I_2$. In this way an output voltage is generated such that

$$I_f = \frac{e_{out}}{R_f}$$

This will continue until $I_1 = I_3$ and $I_4 = I_f$ at which time the diode is back biased and a further increase in e_i will cause no further change in output voltage.

With CR1 biased on, point A will be at a potential of approximately 0.6 volt below ground and this point will remain at the potential until the conduction of the diode is cut off by back bias. If we assume the above statement is true, then the diode can be considered as a constant voltage source with an internal resistance and the gain can be expressed as:

$$G = \frac{e_{out}}{e_{in}} \approx \frac{R_f}{R_1 + R_d}$$

where R_d is the dynamic resistance of the diode.

The above gain equation is sufficient for a reasonable approximation of the gain of the circuit while the diode is forward biased and a more detailed analysis will not be presented.

Therefore, with R_1 determined by the required gain, the break point can be determined as follows:

$$e_{in} = e_{break} \text{ when } I_1 = I_3 \text{ and } I_2 = 0$$

Thus:

$$e_{inb} = I_b [R_1 + R_3] - V_1$$

$$I_b R_3 = e_{inb} + V_1 - I_b R_1$$

assume that for $I_2 = 0$:

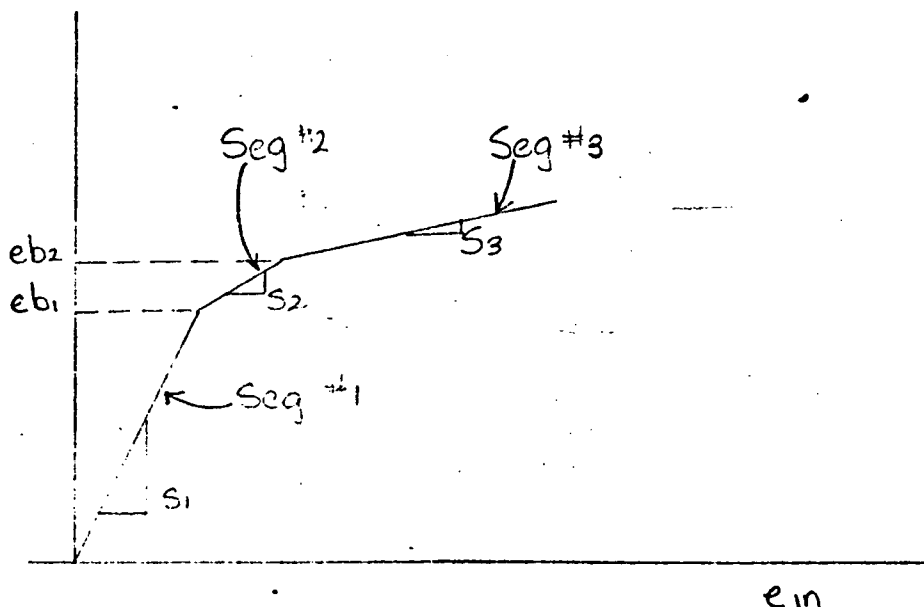
$$e_{inb} = I_d R_1$$

then:

$$R_3 = \frac{e_{inb} + V_1 - I_b R_1}{\frac{e_{inb}}{R_1}}$$

$$R_3 = \frac{V_1}{e_{inb}} R_1$$

Using the equations just derived, the circuit parameters for the three line segments shown below can be calculated:



There are four different transfer functions provided. These are:

1. Linear, $e_{out} = e_{in}$ (Function A).
2. A transfer function where segment #1 has a greater slope than the slope of the equal perceptibility curve (Function B). $S_1 \approx 10$, $S_2 \approx 1$ and $S_3 \approx 1/3$.
3. A transfer function where segment #1 has a slope that is less than the slope of the equal perceptibility curve (Function C). $S_1 \approx 4$, $S_2 \approx 1$ and $S_3 \approx 1/3$.
4. A two segment transfer function selected to complement the camera nonlinearity, $S_1 \approx 2$, $S_2 \approx 0.8$.

From Drawing 528730 the components that determine the different slopes are contained on plug-ins and are: R9, R10, R11, R12, R13, and R14. Variable resistors R27, R34, and R35 are used for the initial setup and for periodic alignments.

It should be understood that the following equations for the break points and slopes are approximations and the actual desired parameters were determined or confirmed empirically.

For Function A:

$$S_1 \approx \frac{R_{16}}{R_9} = 1$$

where:

$$R_{10} = \infty$$

$$R_{11} = \infty$$

$$R_{12} = \infty$$

$$R_{13} = \alpha$$

$$R_{14} = \infty$$

For Functions B and C:

Assume diode resistance $R_d \approx 40$ ohms

$$R_a = \frac{(R_9)(R_{12} + R_d)}{R_9 + R_{12} + R_d}$$

$$R_b = \frac{(R_a)(R_{13} + R_d)}{(R_a + R_{13} + R_d)}$$

then:

$$S_1 = \frac{R_{16}}{R_b} \quad (\text{Slope of Segment No. 1})$$

$$S_2 = \frac{R_{16}}{R_a} \quad (\text{Slope of Segment No. 2})$$

$$S_3 = \frac{R_{16}}{R_9} \quad (\text{Slope of Segment No. 3})$$

$$e_{b1} = \frac{3.3 R_{16}}{R_{14} + R_{35}} \quad (\text{Voltage for break point No. 1})$$

$$e_{b2} = \frac{3.3 R_{16}}{R_{11} + R_{34}} \quad (\text{Voltage for break point No. 2})$$

The diode bridge comprised of diodes CR7, 8, 10, and 9 is turned on during the composite sync time. This allows the capacitor C20 to be charged to a positive level equal to the peak voltage of the sync pulse contained in the video signal. The voltage is positive because the video signal has been inverted by AR1. Utilizing Amplifier AR5, this positive voltage is compared with a negative voltage determined by the position of R20. Thus, the delta voltage is added to the video waveform by AR2. In this manner, the sync pulse is clamped to a voltage level determined by R20. Comparator U1 shapes the sync pulse and transistors Q2 and Q3 supply the necessary current to drive the transformer T1 to turn the diode bridge on and off.

The resistive network, comprised of R37 and R40, establishes a voltage level (1.65 volts) for the comparator to reference the input sync with and change levels if the sync pulse is of greater amplitude than the reference level. Resistors R30 and R36 cause the comparator to have hysteresis effect, thus preventing oscillation when the comparator changes state.

Zener diode CR5 establishes a voltage reference for Amplifier AR6 and the current drive Q1. This voltage reference is used to set the levels for the line segment break points.

A2-2.3 Analog-to-Digital Converter - A1

The Analog-to-Digital Converter is manufactured by Inter-Computer Electronics, Inc., Model IAD-1206. A list of the primary specifications are provided below. For this application, the sample and hold has been bypassed to allow operation at variable sampling rates.

1. Input Voltage	0 to 5 volt full scale
2. Input Impedance	50 ohms nominal
3. Conversion Rate	6 bit binary conversion at ten megahertz rate from internal clock
4. Overall Accuracy	$\pm 1\%$ of FS $\pm 1/2$ least significant bit
5. Linearity	± 1 least significant bit
6. Digital Output	Parallel output straight binary. Logic one "1" is 2.4 (full load) to 5 volts (unloaded). Logic zero (0) is 0 volts ± 0.5 volt. TTL compatible drive capability.
7. Convert Complete Pulse	Pulse is a logic "one" 20 nanoseconds wide that will be available 20 nanoseconds after conversion is completed.

8. External Convert Command

For synchronous drive from an external signal source, pulse should be greater than 4 volts with a minimum width of 20 nanoseconds and a maximum width of 50 nanoseconds. Rise time 10 nanoseconds. Fall time 10 nanoseconds. Nominal input impedance 50 ohms. Conversion triggerable from an external source at rates from DC to 10 megahertz.

9. Environmental

Temperature 0° to 50° C

10. Power

115 VAC \pm 10% 60 Hz
45 watts

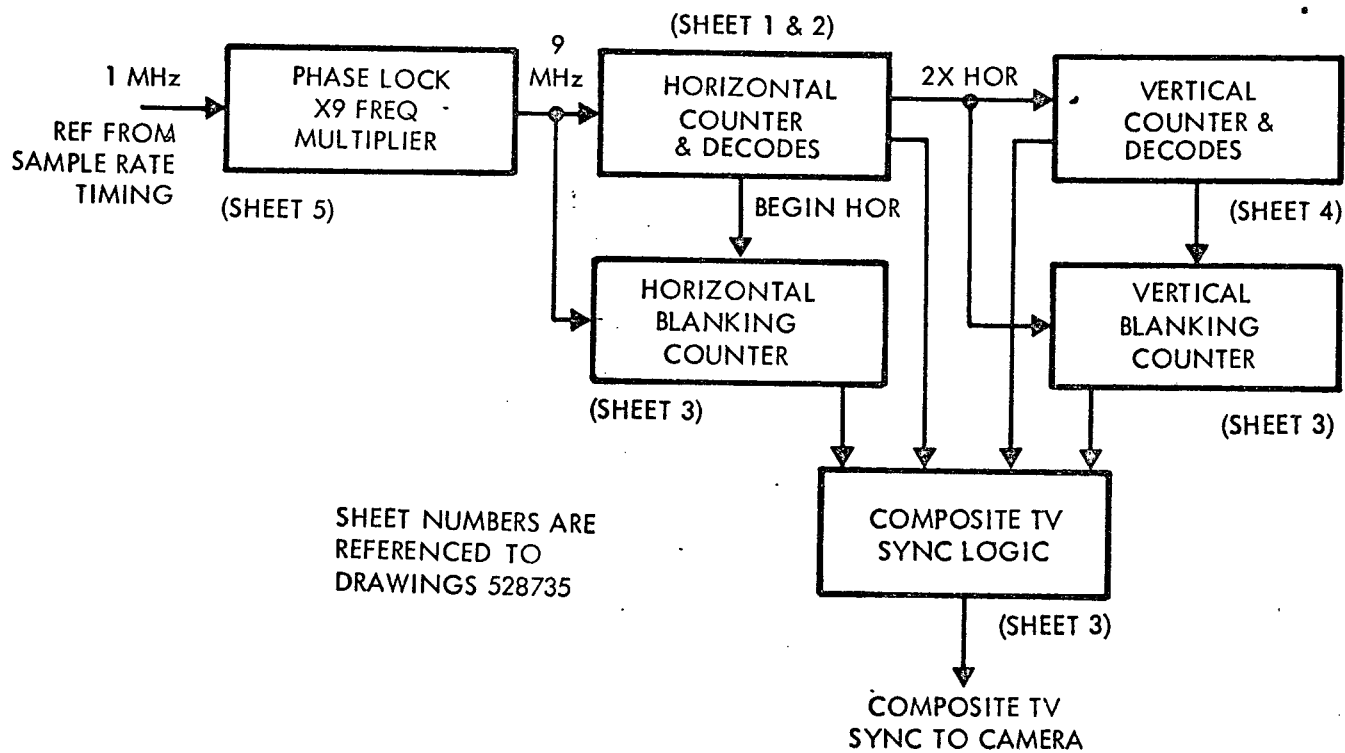
A2-2.4 Sync Separator - A3A2A1

The function of the sync separator is to generate a composite TV synchronization signal to synchronize the TV camera to the system timing. The Sync Separator also provides a blanking signal to control the operation of the Reducer.

A block diagram of the Sync Separator is shown in Figure A2-2.4. The timing is derived from a one Megahertz reference signal from the Sample Rate timing circuitry. This signal is frequency multiplied by a phase lock multiplier to produce a 9 Megahertz signal which is coherent with the sample rate. This 9 Megahertz is used by the Horizontal Counter and Decodes to produce a twice horizontal rate signal, a "begin horizontal blanking" signal, and the horizontally related components of the composite sync signal. Horizontal blanking is generated by a separate counter, from the 9 Megahertz and the "begin horizontal" signal. In a similar manner, the vertically related timing signals are produced from the twice horizontal rate by the Vertical Counter, Decodes and the Vertical Blanking Counter. The Composite TV Sync Logic combines the various horizontal and vertical timing signals to produce the output Composite TV Sync Signal.

A2-2.5 Reducer - A3A1A3

There are two redundancy reduction techniques implemented in the reducer. These are the Zero-Order-Predictor (ZOP) and the Zero-Order-Interpolator (ZOI). The reducer subsystem will process the 6 bit A/D parallel output at a rate (1 to 9 MHz) selected from a thumb-wheel on the front panel. The reducer tolerance will be a four bit number that can be selected as a fixed value or rate-adaptive. In the rate-adaptive tolerance mode, tolerance is controlled by the transmission buffer fill rate.



86185-3

Figure A2-2.4. Sync Separator

Flexibility of parameter choice is supplied by allowing transmission of a 6 bit sample and 4 bit time tag, a 5 bit sample and 5 bit time tag, or a 4 bit sample and 6 bit time tag. These combinations restrict the transmitted nonredundant word to 10 bits allowing simplification of the multiplexing scheme. When 4 bit timing is selected, the 6 bit nonredundant samples chosen by the reducer are transmitted as is. A choice of 5 bit timing causes the first 5 MSB of the nonredundant sample to be combined with the time word for buffer insertion. This is accomplished by gating the MSB of the 5 bit time count into the position normally occupied by the LSB of the 6 bit samples. The 6 bit time word, in a similar manner, combines the first 4 MSB of the nonredundant sample with the 6 bit time for buffer insertion. In this case gating on the first two MSB of the time count is required. Gating is controlled by the sample/time word size selection on the front panel.

Both reduction algorithms require arithmetic operations and various storage elements to perform a reduction operation. Due to the high speed required (9 MHz sample rate) it was necessary to limit a reduction operation to an arithmetic operation (add or subtract) followed by a comparison. The reducer is therefore constructed to perform these operations in such a manner that both reduction techniques use the same circuitry and associated storage elements with only minor changes in register gating (ZOP - ZOI switch in Figure A2-2.5-1) required to differentiate between ZOI and ZOP modes.

In the ZOI Mode, reducer logic operation satisfies the criteria for redundancy removal shown in Figure A2-2.5-2, a flow diagram of ZOI/ZOP operation, and Figure A2-2.5-3, an illustration of ZOI reduction and reconstruction. ZOI operation is accomplished in Figure A2-2.5.1 when the ZOI/ZOP switch is in the ZOI state. This allows AD_1 to be gated to U when necessary and AD_3 to be gated to L when necessary. As shown, the occurrence of a nonredundant sample requires that a new tolerance corridor be established by drawing horizontal lines (U and L) through the limits of the tolerance range placed around the nonredundant sample. This is accomplished by gating AD_1 into U and AD_3 into L after a nonredundant sample is chosen. If a tolerance range (AD_1 and AD_3) corridor placed around a subsequent sample (P_n) overlaps the previous corridor (comparisons made in adders 4-7), P_n is considered redundant and is discarded and t_n (the time tag) is incremented by one in preparation for the next sample check. Each successive redundant sample modifies the corridor extended to the next sample. The new corridor consists of that part of the previous corridor which is overlapped by the tolerance range placed about the redundant sample. In other words, the upper range, either AD_1 or U, and the lower range, either AD_3 or L, forming the most restrictive corridor are saved. If the tolerance range placed about a subsequent sample does not overlap the corridor extended to the sample, this sample is considered nonredundant, and the mid-point of the previous corridor, $(U+L)/2$, is transmitted along with a time tag t_n showing the number of times $(U+L)/2$ must be repeated in the reconstructor. If no nonredundant sample occurs for 14 sample periods (4 bit time), 30 sample periods (5 bit time), or 62 sample periods (6 bit time), the present sample being checked is considered nonredundant.

In the ZOP mode, reducer operation satisfies the redundancy removal criteria shown in Figure A2-2.5-2 with the ZOI/ZOP switches in the ZOP state, and Figure A2-2.5-4, an illustration of ZOP reduction and reconstruction. The basic ZOP operation sets tolerance ranges about a nonredundant sample and then checks successive samples to determine if they are within the corridor formed by the ranges mentioned above. When a sample falls outside this corridor it

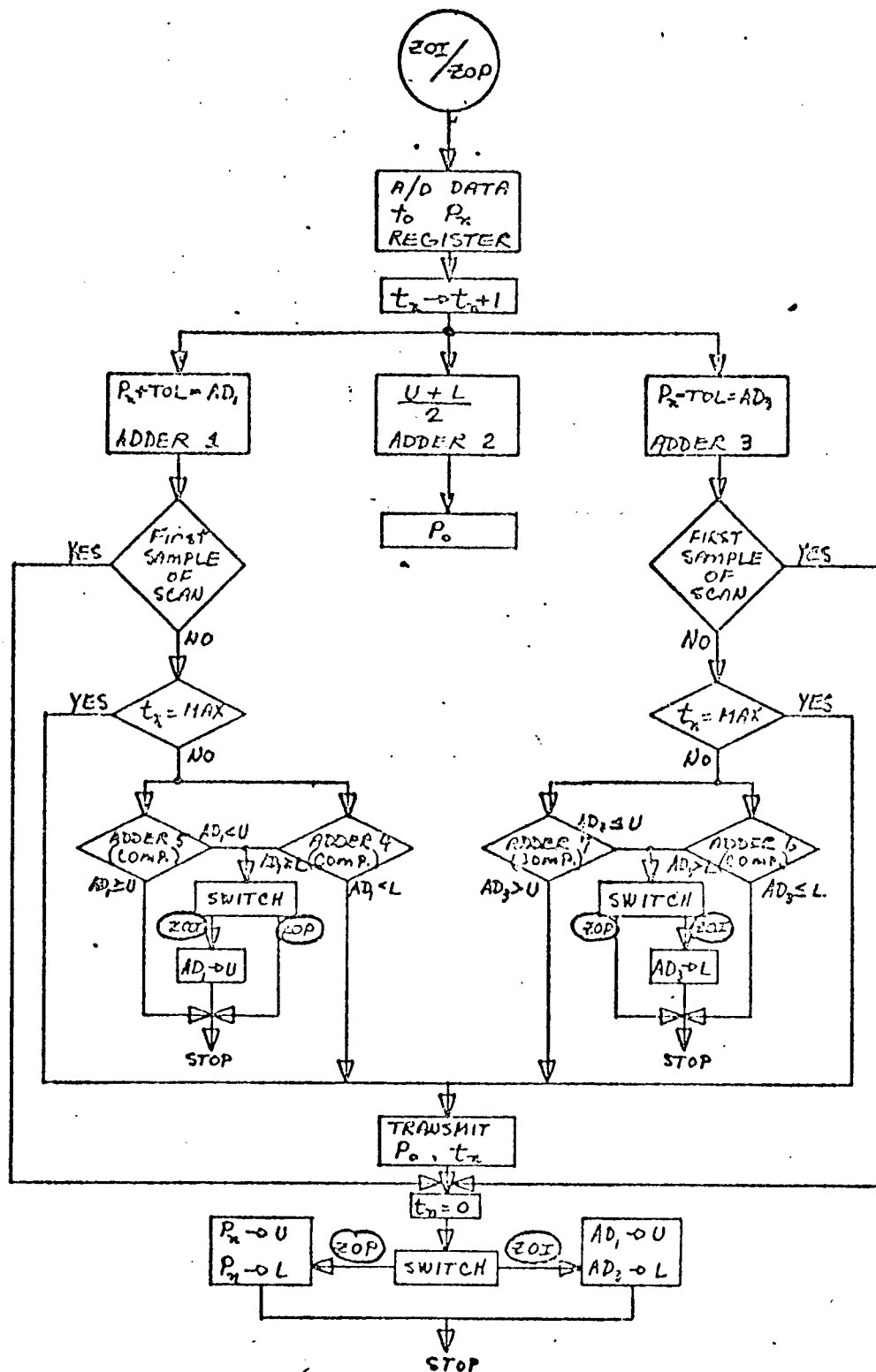


Figure A2-2.5-2. Reducer Flow Diagram

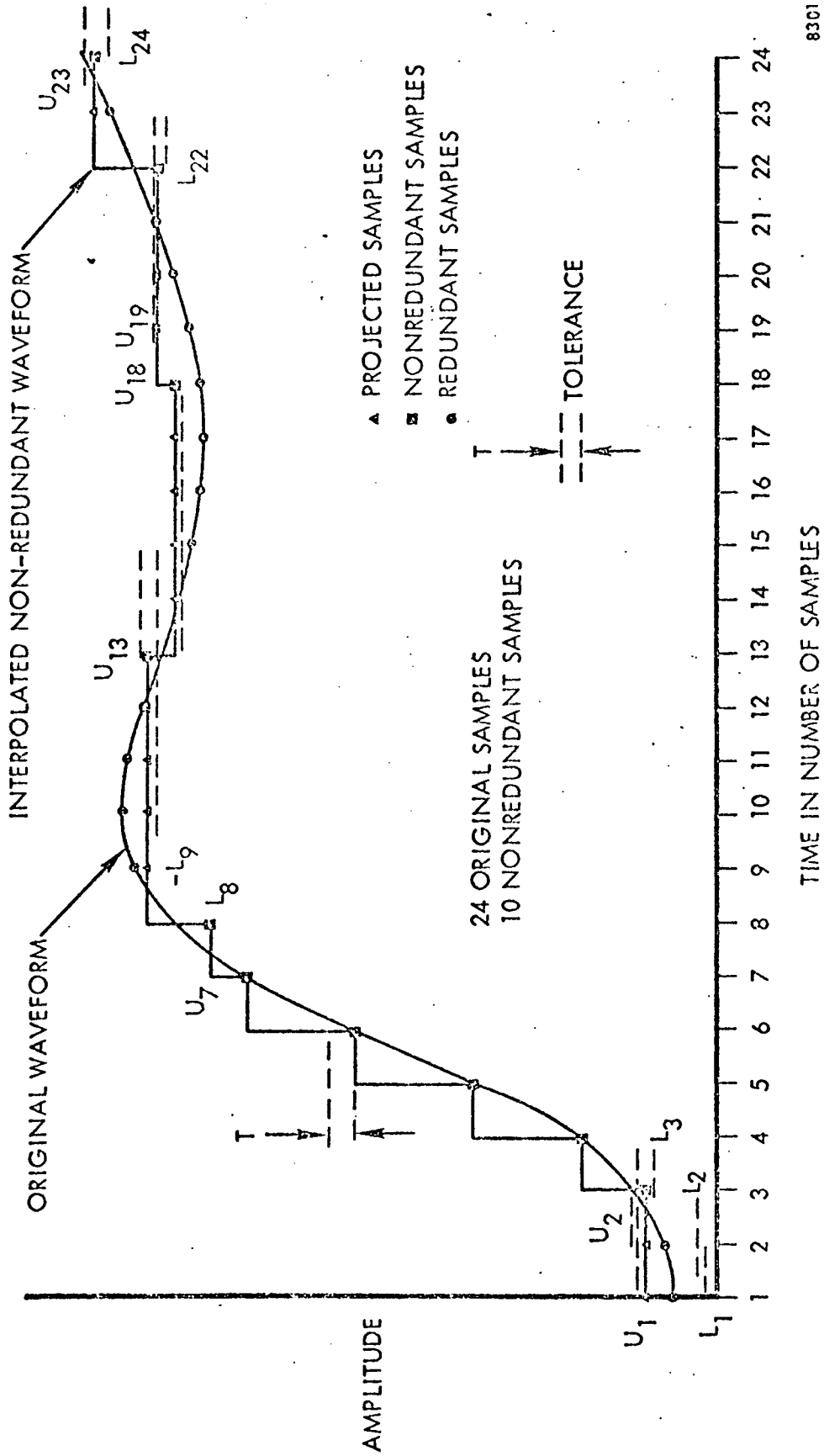


Figure A2-2.5-3. ZOI Redundance Reduction Method

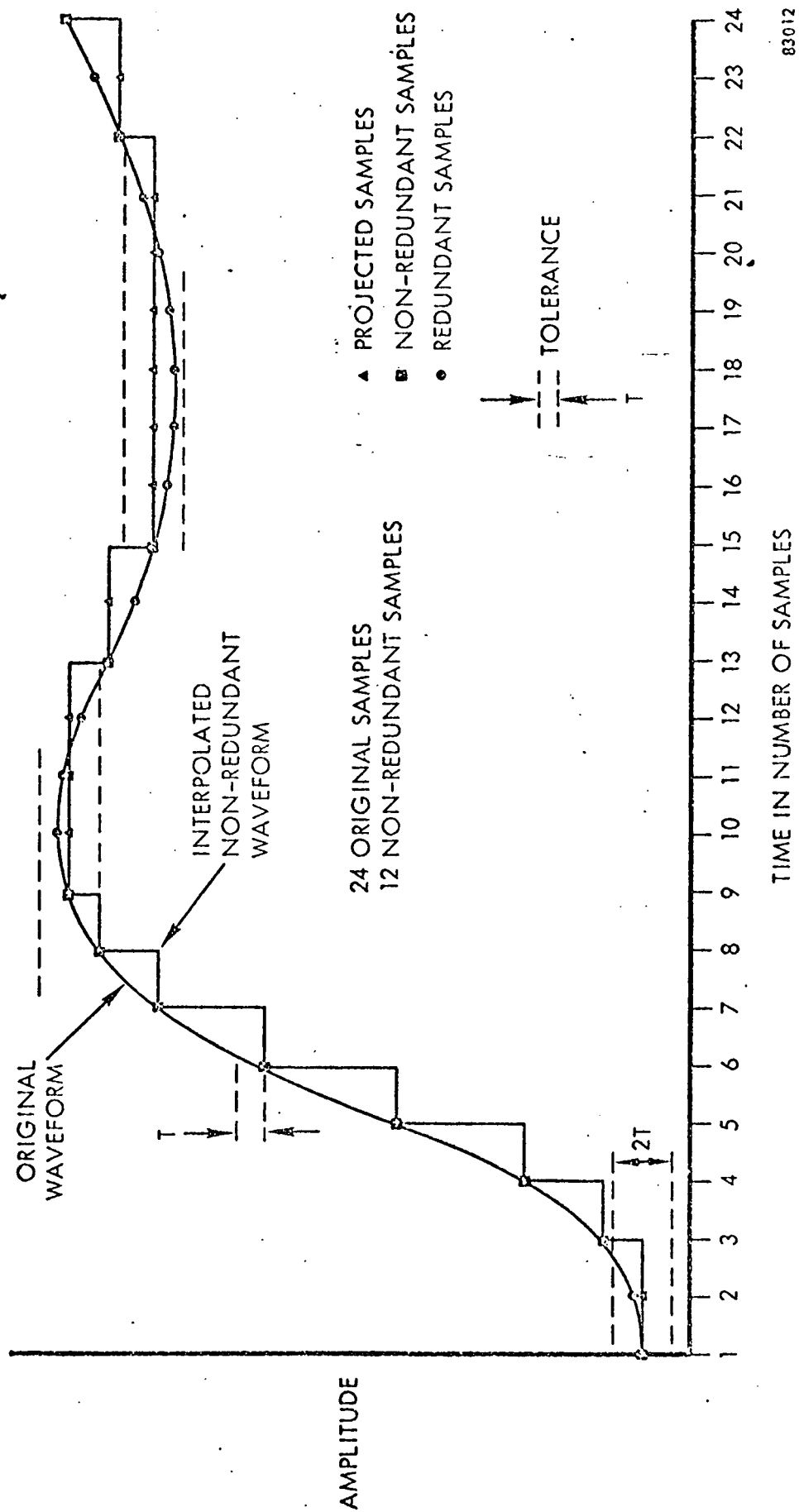


Figure A2-2.5-4. ZOP Redundancy Method

is considered nonredundant and the previous nonredundant sample that formed the corridor is transmitted along with the time tag. ZOP operation in the test set accomplishes the same result but in a different manner. In order to simplify sequencing and control, ZOP reduction occurs by performing exactly the same operations as in ZOI except P_n instead of AD_1 and AD_3 will be gated into both U and L after every nonredundant sample decision (see Figures, A2-2.5-1 and A2-2.5-2 with the ZOI/ZOP switch in the ZOP state). A tolerance range is placed around successive samples (AD_1 and AD_3) forming a corridor which is compared to the previous corridor formed by U and L, where both are equal to the starting nonredundant sample. This check is identical to placing the tolerance range about the starting sample. P_n is considered redundant if the corridors overlap (U and L inside the new corridor), but U and L are not updated by AD_1 and AD_3 due to the gating of the ZOI/ZOP switch. If the AD_1 and AD_3 corridor does not overlap the U-L corridor (P_o), P_n is considered nonredundant, and the previous nonredundant sample $(U+L)/2$ is transmitted along with a time tag showing the number of times it should be repeated in the reconstructor. The sample size-time size choice is identical to ZOI operation.

During TV blanking, when sync information is present in the TV signal, the reducer receives a blanking signal from the sync separator causing the present reduction process to be completed by forcing P_n to be nonredundant and then suspending reducer operation until the blanking pulse is removed. During blanking the only data placed in the transmitter buffer is a special sync code which tells the reconstructor what type of TV sync should occur (horizontal or vertical) at that point in the signal and a buffer sync word which is not presently used. The particular sync word is determined by the type of blanking that is present. When the transmitter buffer is ready to accept data it signals the reducer to insert the proper TV sync word and operates the loading process with its own clock. After TV sync loading is completed the buffer sync word is loaded into memory. The words for horizontal and vertical sync information are made up of a 6 bit zero "time tag" to distinguish them from TV video data and a 4 bit amplitude. This format makes the sync words independent of sample-time word size.

Buffer underflow during TV data processing results in a signal being sent to the reducer forcing it to consider the present sample being processed as nonredundant and inserting it into the buffer. A special operation mode occurs at the resumption of reducer operation after blanking. In order to eliminate the transmission of incorrect data at the beginning of each active TV scan a flip-flop is used to detect the first sample of a TV scan. The normal reducer operation will be bypassed and the U and L registers will be set up with the proper data depending on the reducer mode. Subsequent samples will be processed in the normal manner.

A2-2.6 Statistical Encoder - A3A5

The Statistical Encoder/Decoder system transmits information dependent upon the statistical nature of the sequential data words and the rate of channel activity (encoder memory usage).

Basically the Encoder generates code words that indicate the value difference (delta) in subsequent data words. These codes are chosen to reduce the total amount of data bits transferred across the PCM link.

The data to be encoded is a six bit binary word equal to the intensity of the TV picture at a point in time. The data is positive true with zero data indicating black.

Sample Rate and Data Rate are selected at the front panel and are chosen dependent upon the picture complexity so that the Transmitter memory does not overflow and for the desired picture quality as determined by Sample Rate.

To maintain a constant serial data flow, automatic quantization varies the rate of memory fill by adjusting the number of significant data bits within a range of 4 to 6 bits. These quantization levels, referred to as QL, are automatic submodes and will be detailed in the block diagram description.

The following detailed description of the Statistical Encoder is referenced to the Encoder Block Diagram (Figure A2-2.6-1) and to the Logic Diagrams 531022 and 531023.

Video data from the A/D is provided to the Encoder in the form of a six bit binary data word each Sample Rate time. This video data word is adjusted for quantization at the Select Gate. Quantization Level (QL) 6 enables all 6 bits. QL5 shift selects the 5 MSB bits down one bit to the 5 LSB outputs and provides a filler '1' bit in location 6. QL4 shift selects the 4 MSB bits down 2 bits and provides '1' filler bits in locations 5 and 6. This shifting of data will reduce the number of video levels transmitted but maintains the more significant picture data. An opposite QL operation in the Statistical Decoder restores the proper bit location.

The Input Holding Register and Register B hold two subsequent data words which shift at Transfer time. The data in these two registers is subtracted from one another with the resultant answer as the difference or Delta of the two words. This Delta is updated each Sample time. The Delta is stored as a six bit word and sign, which is presented to the Code Word Select generator.

The Code Word selector compares the Delta word and selects the next code to be generated. The various codes available are:

Delta 0	provides no change in information
Delta +1	provides increase of 1 bit
Delta +2	provides increase of 2 bits
Delta +3	provides increase of 3 bits
Delta -1	provides decrease of 1 bit
Delta -2	provides decrease of 2 bits
Delta -3	provides decrease of 3 bits

Remainder transmits four MSB bits as new data.

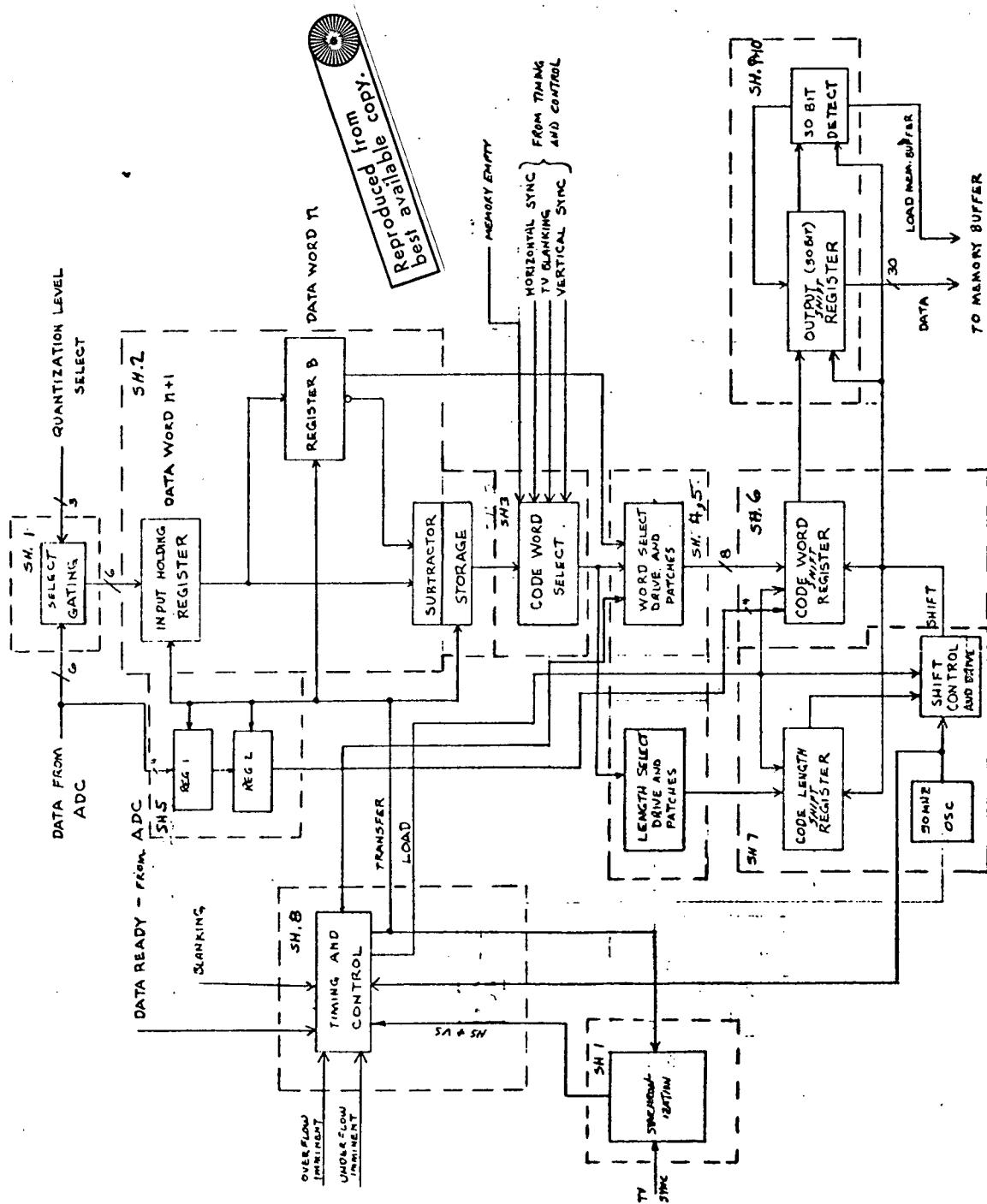


Figure A2-2.6-1. Statistical Encoder Block Diagram

The Word Selector is inhibited during TV Sync and Blanking time and is controlled to some extent when the memory is nearly empty by forcing all Remainder codes during this time, which provides the maximum memory fill rate.

The output of the Code Word Selector is buffered and provided to the Word and Length Patch Selectors. The code word and its bit length is variable by means of plug-in patch modules. Only one code type is active at any one time and is patched to the Code Length and Code Word Registers. These registers are enabled each Transfer time.

The Code Length register operates directly with the Shift Control to produce a discrete amount of clock pulses after each Transfer time. Each Sample Rate produces a Transfer pulse which in turn enables a Load pulse which initiates the Code Generation sequence. The Load pulse loads the current code word information into the Code Length shift register. This code word information is a single '1' bit placed into one of eight stages. All other stages are cleared. The Load pulse also sets the Start flip-flop and inhibits the 50 MHz Clock until the trailing end of the Load pulse. At the end of the Load pulse the clock is released to shift the register until the '1' bit reaches the start flip-flop to turn off the clock. This permits generation of one to eight clock pulses depending on the Code Word length.

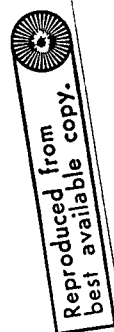
The Shift Control also contains buffer drivers to provide clocks to the Code Word and Output Registers.

The Code Word register is loaded at the same time as the Code Length register but with data information instead of length information. This data is programmable by means of plug-in patch modules or may be actual data in the case of Remainder or Absolute Value code words. The Code Word register is shifted from the Shift Control clock to produce a code data word each Sample Rate.

The Output Shift Register accumulates Data Code Words from the Code Word Register until it has thirty bits of data. At this time, it parallel transfers the thirty bit word into a memory buffer register. The Output Shift register is shifted from the Shift Control clock which is a burst of one to eight clock pulses at a 50 MHz rate.

Parallel transfer of the 30 bit word is accomplished by setting bit 0 of the register to a '1' bit. When this tag bit arrives at the end of the register, it creates a "Load Memory Buffer" pulse to the Memory Interface Buffer. This Load Buffer pulse also clears the Output Shift register and sets the tag bit 0 for the next 30 bit word.

The Memory Interface Buffer (see Figure A2-2.6-2) receives the 30 bit word and sets the Load 30 flip-flop to request a memory Write access of the Memory Control Circuitry. The Memory Interface Buffer provides statistical data to the Data Select gates of the Memory Interface. The other input is ZOP/I data and is selected by a front panel mode select switch. Buffer Drivers are used to interface the 30 bit data word with the memory input.



TV Sync signals are provided from the Transmitter Timing and are synchronized to the Encoder Timing using the Transfer rate signal.

At sample rates of 1, 2, 3 and 4 MHz, Encoder Timing is controlled by Sample Rate which is synchronized to the 50 MHz oscillator and creates a 'Ready Load' and a 'Transfer' timing pulses using a Shift Register. At 5 MHz Sample Rate, to avoid synchronization problems, the sample rate is generated by dividing the 50 MHz oscillator by 10, and routing this 5 MHz to the sample rate circuitry for use in the Statistical Mode only.

TV Synchronization begins at the start of the Vertical Blanking time. The TV Sync signals received are Vertical and Horizontal Blanking signals. These blanking signals are used to inhibit the generation of code data words for the duration of Blanking.

At the end of Vertical Blanking, the Timing and Control section starts a shift register at Transfer rate. This shift register controls the generation of four code words used to control the Decoder start time. The first word is a 30 bit Time Code Word generated in four segments. This code is a fixed pattern of

000001000000010000000100000001

The next three code words are eight bits each and are the Vertical Sync code, Buffer Word and the Absolute Value word.

The Vertical Sync code is used by the Decoder to start the decoding process. The Buffer word is a spare 8-bit word of all zeros. The Absolute Value word is comprised of two bits of Quantization Level information indicating level 4, 5, 6 to be used for the current horizontal line only. The remaining six bits are a real data value used by the Decoder as a video level starting point.

Horizontal Synchronization occurs at the end of Horizontal Blanking and generates three 8-bit code words which are the Horizontal Sync code, Buffer word, and Absolute Value word identical to those of the Vertical Sync time with the exception of the bit code of the Horizontal Sync word.

The Operation of the Timing and Control is as follows: The T & C receives four timing signals, 1) A 50 MHz clock, 1) Sample Rate, 3) Horizontal Sync (blanking), 4) Vertical Sync (blanking).

As each Sample Rate is received, it is synchronized to the 50 MHz internal clock and placed in a shift register to generate two timing pulses, the Ready Load and the Transfer. The Ready Load is used to setup and strobe the T & C logic and to generate the Load Command to the code generator. The Transfer pulse occurs immediately after the Ready Load and causes the transfer of data in the various parallel registers to update for the next sample rate time. These two pulses operate continuously at the Sample Rate.

Vertical and Horizontal Sync signals inhibit the generation of code words for the duration of either signal (that is, the blanking time). This is accomplished by a second shift register which synchronously inhibits Load pulses following the first Load pulse after the start of blanking. Upon the end of blanking, either Horizontal or Vertical, the Load pulses are enabled to start the next video line. The first words of the Video line are the Time Word and sync words in the case of Vertical sync or just the three sync words in the case of Horizontal sync. The remainder of the line generates a code data word for each sample rate until blanking occurs.

Automatic Quantization avoids uncontrolled overflow or underflow of the Transmitter Buffer. Since the Code Word is variable length, one bit to eight bits, the 30 bit word rate out of the Encoder to the buffer varies. The average input and output of the buffer must be equal and therefore it is necessary to provide feedback control of the encoder.

The Quantization is controlled by the Rate Buffer and is readjusted each Horizontal Word time. The Buffer fill rate is determined by an Up/Down Counter which contains the current fill rate at the end of each Horizontal line. This rate may be positive or negative since the counter is preset to 128. The resultant count is compared to two fixed numbers each of which may be one of two values selected by the Memory fill function.

The two numbers are compared against the Rate Buffer counter and fixes the QL of the next line at one of three values depending whether the rate is positive, negative, or average. The three QL values are 4, 5, and 6. The Video resolution is reduced from 6 to 5 or 4 bits as the QL goes to 5 or 4. A low rate selects QL6, an average rate selects QL5, and a high positive rate selects QL4 to reduce the rate of Buffer fill.

Once the Quantization level has been selected for the next horizontal line, the QL shifts the input Video data down zero, one, or two bits creating a six, five, or four bit resolution. The lower the resolution, the fewer large picture value changes take place and more 'no change' or one to three bit changes occur. This transfers the work load from 8-bit Remainder codes to one-bit Delta zero and three-bit Delta one codes which cause a lower fill rate.

The shifting of the Video data words to 5 or 4 bits is compensated for in the Decoder by storing the current QL value of the Absolute Value word and reverse shifting the reconstructed Video data word to produce a six-bit output of the Decoder.

The Transmitter Buffer memory has a buffer empty signal (four words left) which forces the Code Word Selector to produce a series of eight-bit Remainder code words to generate a positive fill function to keep the memory from becoming empty. This may happen at any time during the code generation time.

A2-2.7 Transmit Rate Buffer Subsystem - A3A1A4

A2-2.7.1 Rate Buffer

The transmit rate buffer accepts data from the data compressor or statistical encoder, assembles it into 30-bit words, and writes the assembled word into memory. It reads 30-bit words and sends them to the multiplexer when required. The buffer subsystem also formats the input TV and buffer synchronization words and calculates the buffer sync word. Blanking fill words are inserted into the multiplexer when required.

A2-2.7.2 Detail Operation

The Transmitter Rate Buffer is a 512-word, 30-bit per word high speed memory. This buffer is based on analysis which shows that a larger buffer does not improve system performance.

TV video has active periods of a specific time duration (active portion of a line) and completely inactive periods also with a specific time duration (blanking). During active periods, data is going into and out of the buffer; thus, buffer fill may increase or decrease. Since only buffer output occurs during blanking, buffer fill will decrease at a rate determined by the transmission rate while horizontal or vertical blanking occurs. Due to the distributed nature of horizontal blanking in a field, buffer fill during a field will roughly resemble a sawtooth waveform at line rate, superimposed on a larger sawtooth at field rate.

TV frame activity changes slowly enough that a sequence of many fields will have approximately equal activity. If significant increase in buffer fill occurs over a complete field (including blanking), any feasible size buffer would overflow before the activity could be expected to decrease. For example, assume that data activity does not change significantly over 100 fields (about 1.7 seconds); the net buffer fill per field can only be 1/100 of the total buffer size. Even for a 3,000-word buffer, this is only 30 words per field. At a transmission rate of 9 megabits there are 15,000 10-bit words per field (9×10^6 bits/sec \times 1/60 sec/field/10 bits/word). Thus, the extra 30 words per field are a small fraction (0.2 percent) of the total words, and would not significantly improve the picture. Therefore, the buffer may as well be sized so that it can be emptied during vertical blanking. For a vertical blanking time of 1250 μ sec and a maximum transmission rate of 9×10^6 bits/sec or 0.9×10^6 words/sec, the maximum buffer size required in the DC/EC Test System is $1250 (0.9 \times 10^6) = 1125$ words. The buffer size provided is $512(3) = 1536$ words. Lower transmission rates have a smaller buffer requirement.

The transmit rate buffer consists of the random access memory and the buffer control logic shown in Figure A2-2.7.2.

The transmit buffer has two sources of input data. Consider first the reception of data from the Reducer. The 10 data lines from that source are updated each sample time. When the data present represents a nonredundant sample, a signal (Flag XX) is sent to the timing and

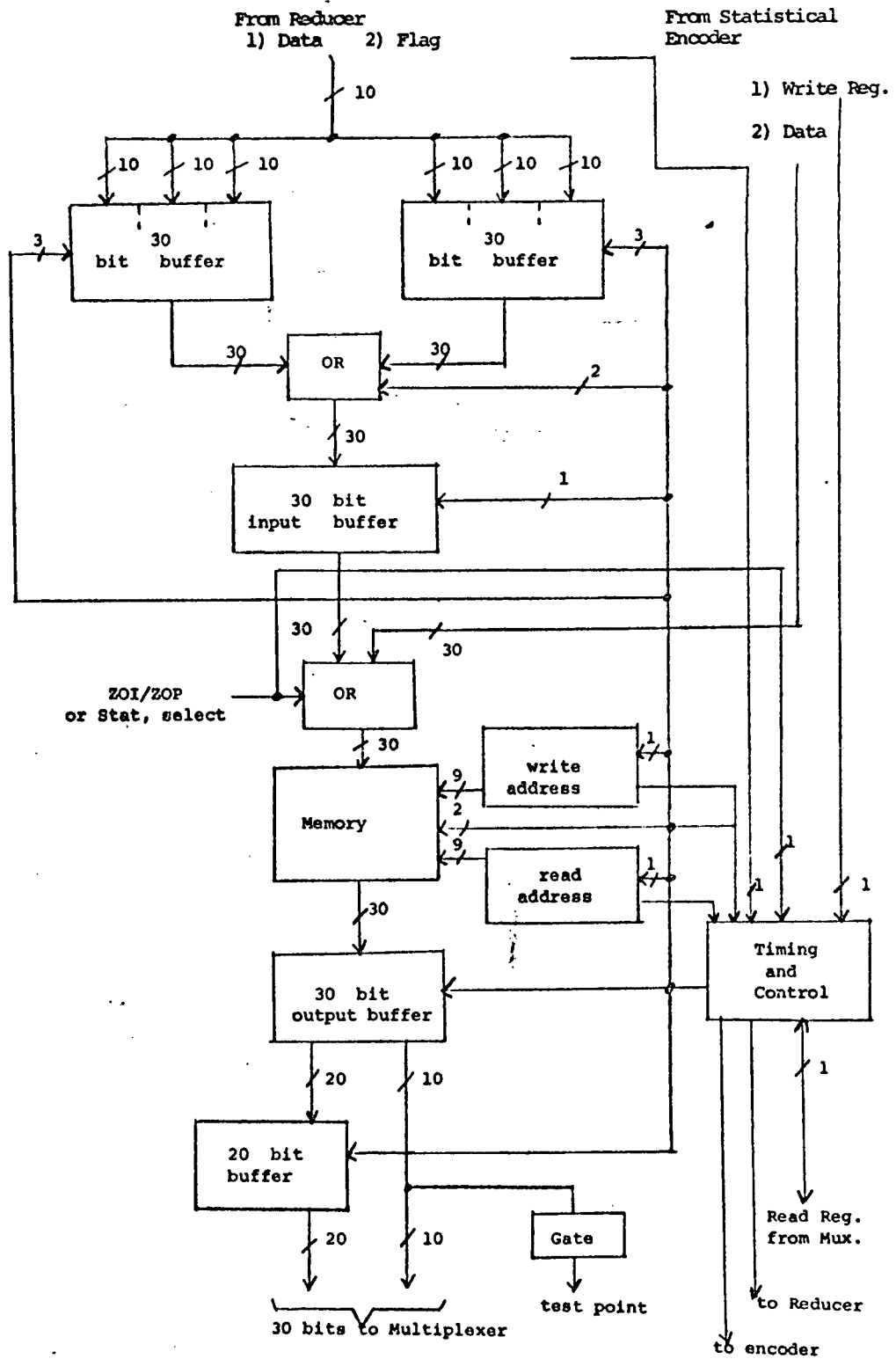


Figure A2-2.7.2. Transmitter Rate Buffer Block Diagram

control (T&C); there it is gated to one of six 10-bit sections in two 30-bit buffers and causes the selected section to accept the data. The same signal then updates a Mod-6 counter in the T&C and qualifies the selection of the next 10-bit section.

The Mod-6 counter is decoded so that when one of the 30-bit buffers is filled, a write request is generated. Coincidentally, unless a prior request is still pending, the data in the filled 30-bit buffer is transferred to the 30-bit input buffer. Under worst-case conditions, 4 of the 6 available 10-bit sections will be filled before a transfer can be made to the 30-bit input buffer. This happens when:

1. The reducer finds nonredundant samples in 8 adjacent sample periods, and
2. The sample rate is 9 MHz, and
3. The write request decoded from the Mod-6 counter occurs at the beginning of a memory read cycle.

Data from the 30-bit input buffer is routed to the statistical decoder logic where it is qualified by a mode selection switch before being gated to the memory.

When the T&C recognizes the write request, the mode line to the memory is set low and a cycle initiate command (MRQ) issued. (Requests for memory service are handled on a priority basis so that if read and write requests occur coincidentally, the write request will be serviced first.) Once the write cycle is completed, the T&C increments the write address counter by 1 count.

If the Statistical Encoder unit is selected, operation is as follows. 30 bits of data are supplied to the memory inputs through a set of gates. Write requests decoded from the Mod-6 counter are inhibited and in their place write requests from the encoder are serviced. Thirty bits from the encoder require more than 3 sample periods to accumulate and the highest encoder sample rate is 5 MHz; so, input buffering at the transmit buffer is unnecessary.

Output service can now be considered. Data readout of the memory is in response to link demands. 30 bits are made available to the demultiplexer (demux). When the first 10 of these are accepted from the 30-bit output buffer, the remaining 20 to be used are transferred from the output buffer to the 20-bit buffer.

The demux signals its acceptance of the first 10 bits when it sends a read request (TV3) to the T&C. While the demux uses the remaining 20 bits, the T&C services the request and loads a new 30-bit word into the output buffer. When the read cycle finishes, the read address counter is incremented by 1 count.

Bits 1 through 10 of the output buffer are decoded in an "and" gate to provide a test point. In the ZOI/ZOP mode the point should sense vertical sync.

Next, consider generation of special words such as buffer sync and TV sync. At the beginning of a horizontal line blanking period, the T&C will command the reducer to output

a horizontal sync word. If one of the two sections of a 30-bit buffer are filled with samples, the T&C will enter a TV sync word in each remaining section and store the data. In the next 30-bit buffer's first two sections, the T&C will enter a TV sync word and then a buffer sync word. This format was selected so that each TV line is represented by some whole number of memory words, and so that the buffer sync word is always located in the middle slot next to a TV sync word. The buffer synchronization scheme utilized in the Receiver buffer does not require this formatting procedure, but no attempt was made to remove the generating logic from the transmit buffer.

The buffer sync word, used in the statistical encoder, is calculated by subtracting the address difference (write-read), from the maximum buffer size selected.

At vertical blanking time the same sequence is used as at horizontal blanking time. The important difference, however, is that the sequence is started at about 1 μ sec before the end of vertical blanking.

Fill words are generated when the memory is empty and TV blanking time is occurring. Generation is carried out in two ways. When the ZOI/ZOP mode is used, the T&C commands the reducer to generate fillwords. These are then passed through memory in the normal manner. In the statistical encoder mode, the T&C answers demux requests by loading fillwords directly in the 30-bit output register.

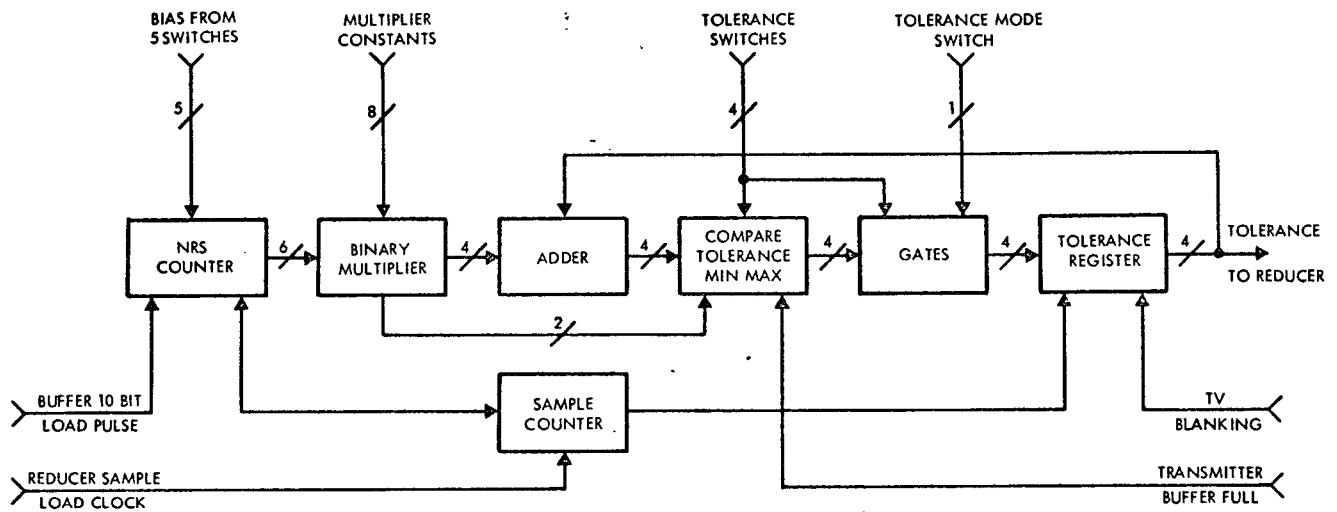
A2-2.8 Rate Adaptive Tolerance Generator - A3A1A1

The tolerance generator produces a 4-bit tolerance that is a function of the actual buffer fill rate and the desired fill rate. A minimum tolerance is switch selectable along with the desired buffer fill rate. A fixed tolerance mode is provided that outputs a switch selectable fixed tolerance to the data compressor.

A2-2.8.1 Detailed Description

The tolerance generator makes use of the average rate of buffer fill in determining tolerance. This provides feedback to the data compressor as a control on buffer fill. This control is accomplished by increasing tolerance when the buffer fill rate increases and decreasing tolerance as fill rate decreases. This control makes use of the general rule that increasing tolerance decreases the average rate of occurrence of nonredundant samples thus decreasing buffer fill rate. Figure A2-2.8.1 is a block diagram of the tolerance generator.

Two counters are used to determine the average rate of buffer fill, to compare this rate to a desired fill rate, and to generate an error signal which is used to modify tolerance. One counter, the NRS counter, starts from a preset negative bias obtained from 5 bias switches, and counts the 10-bit words that are loaded into the transmitter buffer. The other 5-bit counter counts the samples that are loaded into the data compressor. Every 32 input samples this last counter interrogates the NRS counter and presets it to the selected bias and another counting cycle begins. The value of the NRS counter when sampled is equal to the difference between



86183-5

Figure A2-2.8.1. Tolerance Generator Block Diagram

the desired number of nonredundant samples (represented by the bias) and the actual number of nonredundant samples that have occurred during 32 input sample times. In other words, the counter sample is a measure of the difference between actual and desired buffer fill rates and is a suitable error signal for the buffer fill rate control loop. The desired buffer fill rate can be shown to be a function of the transmitted information rate and the input sample rate. The bias is thus programmed as a function of these two parameters.

The NRS counter value observed at the end of each 32 sample period is processed by a binary multiplier. The switch selectable multiplication factors are 4, 2, 1, 1/2, and 1/4 and may be selected independently for positive and negative NRS counter results. The resulting number is used as a tolerance modifier (ΔTOL). ΔTOL is the error signal of the control loop. If it is positive, the actual buffer fill rate is greater than the desired fill rate, and tolerance is increased in an attempt to reduce the actual fill rate. Tolerance reduction results when the actual fill rate is less than the desired value. This is accomplished by adding ΔTOL to the previous tolerance which is stored in the 4-bit tolerance register.

The new tolerance is now compared with the minimum and maximum tolerance values. Maximum tolerance is 15 due to 4-bit tolerance. Minimum tolerance, which is required due to noise on the input A/D signal, is determined by 4 tolerance switches. When the transmitter buffer fills up, the nonredundant sample rate changes to the transmitted 10-bit word rate. Since this rate may force the tolerance generator to reduce tolerance at a time when tolerance should be large, the buffer full signal from the transmitter buffer is used to force tolerance to its maximum value.

The 4 tolerance switches mentioned above are used to set fixed tolerance when the tolerance generator is in the fixed tolerance mode. In this mode rate adaptive tolerance continues to be generated, but the tolerance switch values are gated into the tolerance register instead of the adaptive tolerance.

A2-2.9 Voice Encoder - A3A3A4

The Voice Encoder digitizes an audio signal to a 20 K bit data stream. Analog amplitude compression is available at the voice encoder input. The digitizing technique is high information or variable slope delta modulation, a method which has been demonstrated to be somewhat superior to other digitizing methods at low bit rates.

The voice compressor consists of an operational amplifier with a diode break type nonlinear feedback circuit. When the voice input is 50 mV, the current in the feedback network is low, and all the diodes are reverse biased. Thus, the equivalent feedback resistance is about 32 K ohms, giving a small signal gain of 30 dB. As the input voltage increases to a maximum of 5 V (60 dB over 50 mV) the diodes successively turn on, giving a small signal gain of 0 dB at 5 V input. Thus, the output range is 30 dB for a 60 dB input range, giving a compression ratio of 2:1. The voice compressor may be bypassed. In the bypass mode, the audio input signal must be less than 5 V peak-to-peak maximum.

The Voice Encoder (Drawing 528729) contains a voice compressor, which compresses the normal 60 dB voice range into 30 dB, and a high information delta modulator, which encodes the voice output into a digital data stream. Input impedance may be selected to be 600 or 50 ohms.

The high information delta modulator has a modulator and a demodulator. The modulator encodes the voice information from the compressor into a serial data stream. The demodulator generates a feedback signal which is used as an error indication in the modulator.

The audio is fed to one input of a comparator, and the demodulated signal from the demodulator is fed to the other input. The comparator operates a transistor driver stage, and is powered by ± 15 volts power regulated to +12 and -6 volts by zener diodes. When the incoming voice signal is larger than the feedback signal, the transistor is switched on and the comparator output is a logic zero. When the voice input is smaller than the demodulated feedback signal, a one is generated.

The comparator output is sampled at a 20 kHz rate on the clock high going transition by a D flip-flop. The flip-flop output drives the input of a 2-bit shift register. Decoding circuitry on the outputs of this shift register operates commands to a 4-bit shift left/shift right (SLSR) register in accordance with the following table.

A follows B follows C

$A \neq B \neq C$	Shift Left
$A \neq B = C$	Shift Left
$A = B \neq C$	Hold
$A = B = C$	Shift Right

The low going transition of the clock strobes the shift left, shift right, and hold commands into two flip-flops. One of these controls the "mode" input of the SLSR, which determines the shift direction. The other flip-flop disables the clock to the SLSR for one bit period when a hold command is present at its input.

The SLSR shifts ones in from the left and zeroes in from the right. Four "exclusive-OR" gates on the SLSR outputs compare the outputs with the comparator output to generate a 5-bit ones complement output code. The "exclusive-OR" gates and comparator output operate transistor drivers. The comparator output, which is used as the sign bit, is buffered for the modulated voice output.

The outputs of the transistor drivers go to a resistor network at the input of the demodulator. The demodulator consists of an op amp integrator. The input network is designed so that when the SLSR shifts right the output slope of the integrator is doubled. When the SLSR shifts left the slope is halved. When the SLSR state is all zeroes, the normalized slope of the

integrator is +1 when the sign bit is a one, and -1 when the sign bit is a zero. Thus, as the SLSR shifts right, slopes achieved will be 1, 2, 4, 8 and 16.

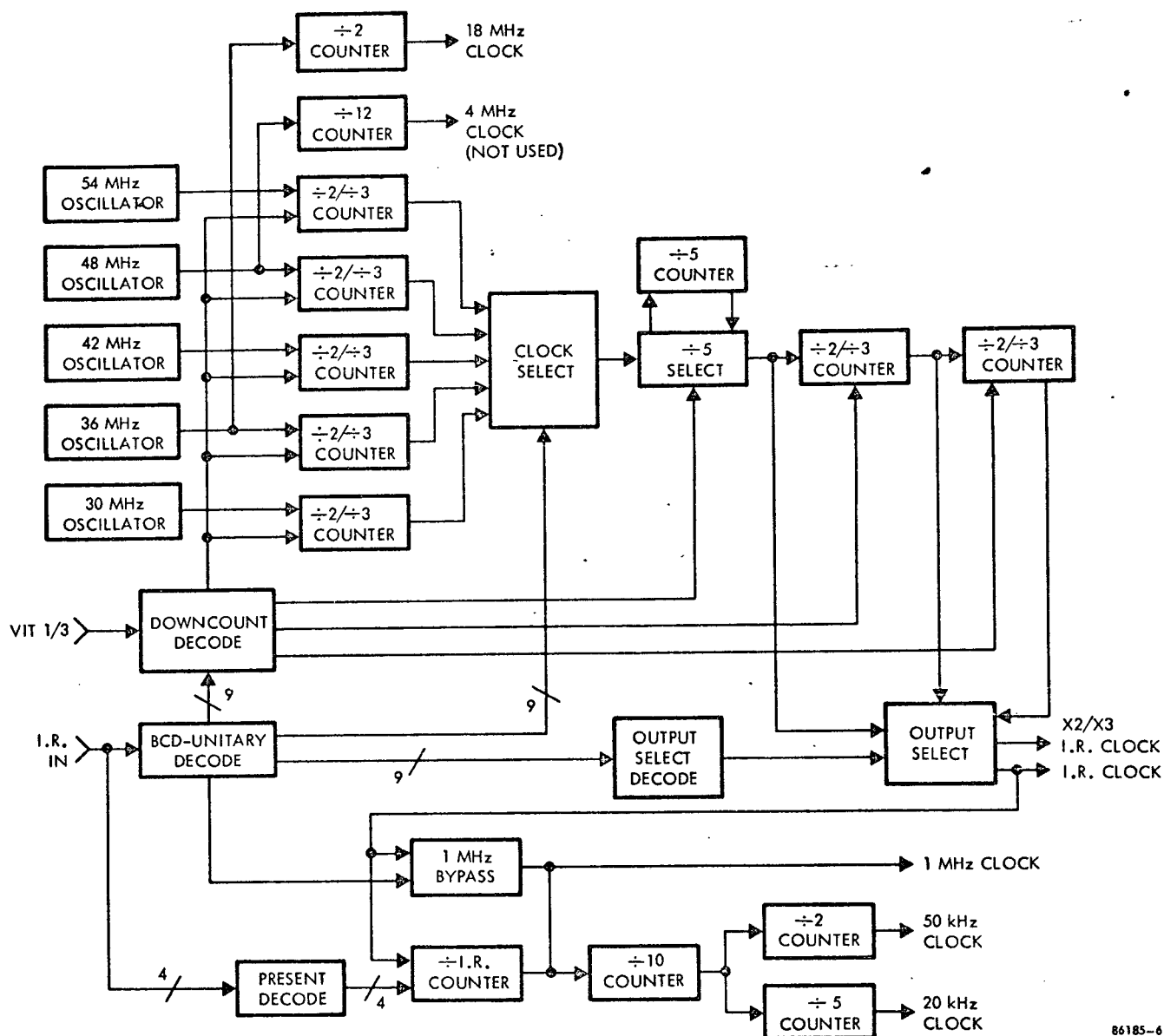
From the table, it is seen that if the previous three bits were identical, indicating a large error between the voice signal and the demodulator output, the SLSR shifts right and the demodulator slope is doubled to help the demodulator "catch up" to the voice input. If the demodulator "catches up" at a time when its slope is large, the sign bit changes, resulting in a polarity reversal of the demodulator slope, and the slope is halved. For step inputs this results in an underdamped characteristic, but for most audio signals, reproduction will be better than with other voice encoding methods.

A2-2.10 Information Rate Timing - A3A2A6

The transmitter information rate timing circuit (Figure A2-2.10) generates clock signals at frequencies of 1 through 9 MHz variable in steps of 1 MHz. Depending upon the coding technique used, a clock (bit rate) of either two times or three times the information rate, synchronous with the information rate, is also generated. Other clocks that are generated for use elsewhere in the system are at frequencies of 18 MHz, 1 MHz, 50 kHz, and 20 kHz. The 18 MHz clock is used in the transmitter rate buffer. The 1 MHz clock is used as a reference to generate the sample rate clock and the coherent 9 MHz for the TV Sync Separator. The 50 kHz clock is used in the multiplexer, while the 20 kHz clock is used by the voice encoders.

The transmitter information rate timing circuit has five digital clocks operating at frequencies of 30 MHz, 36 MHz, 42 MHz, 48 MHz, and 54 MHz. These clocks are divided down to the desired information rate, and two times or three times information rate frequencies by means of a programmable counter circuit and appropriate switching circuitry. The positions of the information rate code switches on the front panel are decoded to determine the mode of operation of the information rate down counter circuitry. A presettable down counter further reduces the information rate to 1 MHz. Additional down counters reduce the 1 MHz signal to 50 kHz and 20 kHz clocks. An 18 MHz clock is produced by dividing the 36 MHz clock oscillator output by two.

The outputs of these clocks are buffered by high speed Shottky logic gates and go immediately into programmable divide-by-two or divide-by-three counters. The output of the divide-by-two/divide-by-three counter is at $1/3$ the frequency of the input signal when the program line is a logic "1." The output is $1/2$ the input frequency when the program line is a logic "0." The clock frequencies are thus immediately divided by either two or three to obtain frequencies which can be more easily handled by the clock switching circuitry. The switching circuitry consists of five Shottky gates with enabling lines and an 8-input high speed gate. The highest frequency that will be seen by the 8-input gate is 27 MHz. Following the clock switching circuitry is a programmable divide-by-5 gate which may be used to obtain some information rate frequencies. After the switchable divide-by-5 circuit, there are two more divide-by-two/divide-by-three circuits, whose outputs go to the switching circuit. It is at the



86185-6

Figure A2-2.10. Transmitter Information Rate Timing Block Diagram

output of these switching circuits that the information rate and two or three times the information rate clocks are taken.

To illustrate the operations of the decoding circuitry and information rate down counters, let us assume it is desired to obtain a 3 MHz information rate with a code 1/3. The information rate switch on the front panel will be set to 3, and the code switch on the front panel will be set to Viterbi 1/3. The VIT 1/3 input of the transmitter information rate timing circuit will be a logic "0," and the IR 2 and IR 1 lines from the information rate switch will also be at ground. The BCD to decimal decoder in the decoding circuitry will interpret the BCD input from the information rate switch and its output in conjunction with the VIT 1/3 line will be decoded to enable the proper circuitry to obtain an information rate of 3 MHz. The clock signals will all be divided by 3. The 54 MHz clock will be selected so that the input to the rest of the down counting circuitry is at a frequency of 18 MHz. The divide-by-5 counter will not be used, and thus the signal will go to the second divide-by-two/divide-by-three counter stage. This counter stage will be set to divide-by-two to obtain a frequency of 9 MHz, and its output will be selected to be the times 3 information rate clock. The final divide-by-two/divide-by-three counter stage will be set to divide by three to obtain a 3 MHz signal. Its output will be selected to be the information rate clock. In a like manner, any other required combination of information rate and two or three times information rate can be generated from one of the clock oscillators.

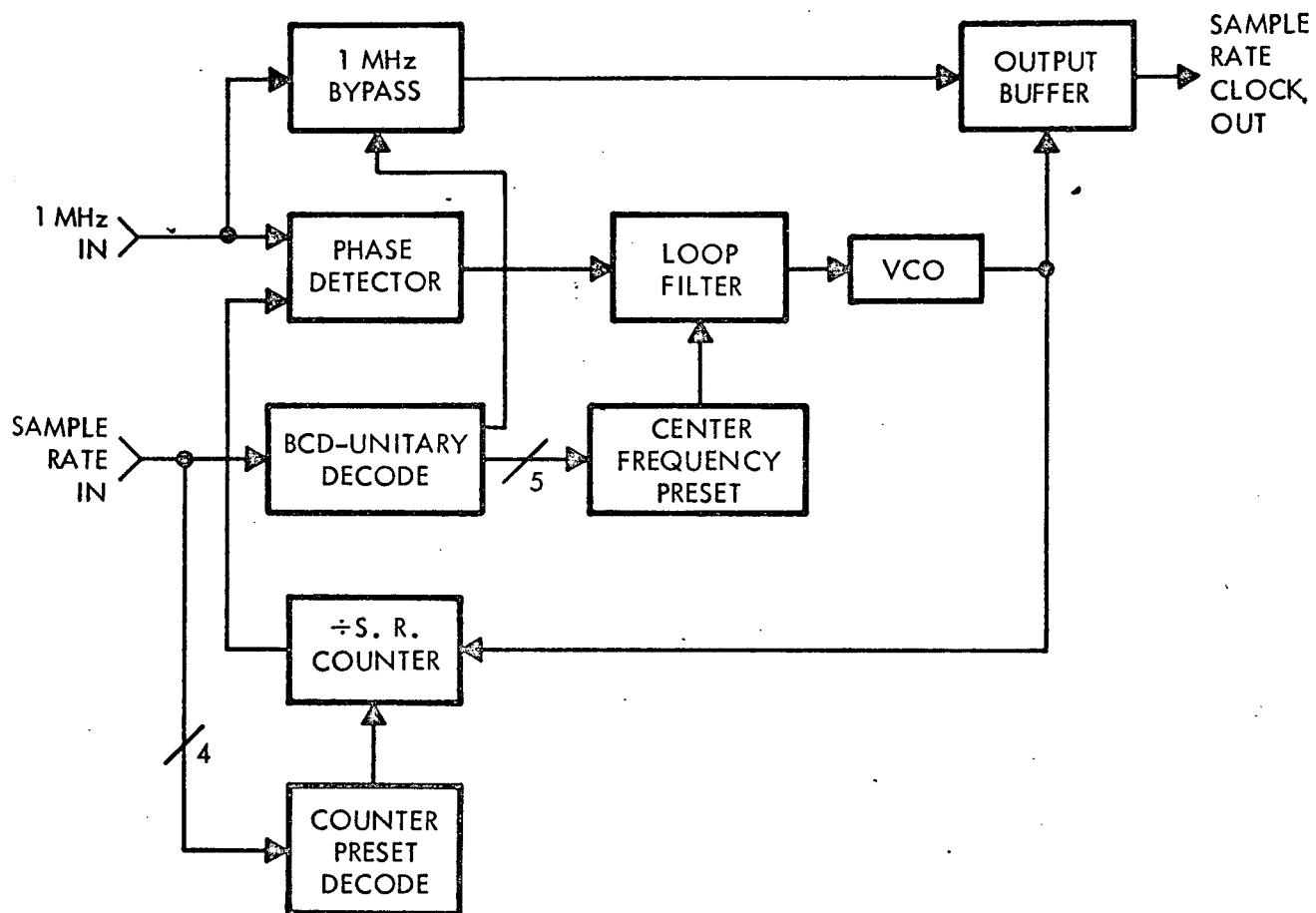
An 18 MHz clock signal is generated for use in the transmitter rate buffer by dividing the output of the 36 MHz clock oscillator by two. This is done by a flip-flop.

To generate the one MHz clock the BCD input from the information rate switch on the front panel is applied to one set of inputs of a 4-bit adder. The other inputs are hardwired to a count of 10. Since the BCD input from the information rate switch is inverted, the output of the 4-bit adder will be 10 minus the information rate. This is applied to the data inputs of a 4-bit synchronous decade counter. The carry output of the counter is connected back to the load input by an inverter. Thus, when the counter reaches its full count, it will load 10 minus the information rate into the counter. Since the counter is run from the information rate signal, the signal at the carry output will be at a frequency equal to the information rate clock divided by the information rate, or one MHz. When the information rate is set to 1 MHz, the synchronous counter is disabled and the information rate is routed directly to the 1 MHz output.

To generate the 50 kHz and 20 kHz clock signals, the 1 MHz clock is first divided by 10 by a decade counter. This generates a 100 kHz clock which is divided by two to obtain the 50 kHz clock and by five to obtain the 20 kHz clock.

A2-2.11 Sample Rate Clock - A3A2A6(A7)

The transmitter sample rate circuitry (Figure A2-2.11) generates a programmable clock from 1 to 9 MHz phase locked with the information rate clock.



86185-7

Figure A2-2.11. Transmitter Sample Rate Timing Block Diagram

To generate the programmable 1-to 9 MHz clock output, a phase-lock loop is employed which phase locks this clock output to the one MHz reference output of the information rate timing circuits. The center frequency of the phase-lock loop is selected by the sample rate selector switch on the front panel.

The one MHz reference signal is applied to a phase detector and is compared with a one MHz feedback signal from the phase-lock loop. Phase lead, or lag, is detected and filtered by a loop filter. The output of the loop filter drives a voltage-controlled oscillator. For a sample rate of one MHz, the one MHz reference input is routed directly to the output.

The sample rate output frequency is divided down to one MHz by a presettable synchronous down counter, and this signal is applied to the phase detector to be compared with the reference input. The center frequency of the voltage-controlled oscillator is selected by varying the dc offset of the loop filter output. This is accomplished by means of five relays, each one of which switches a different offset voltage at the output of the loop filter. The relays are selected by a decoding circuit.

The synchronous upcounter is preset to a count which is equal to 10 minus the sample rate selected on the front panel switch. This number is derived by adding the inverse of the sample rate in BCD to a fixed count of 10. This is accomplished using a 4-bit binary full adder. The outputs of the binary full adder are applied to the data inputs of the synchronous upcounter. When the upcounter reaches a full count, its carry output is routed to its load input so that the number generated by the binary full adder is loaded into the counter on the next clock pulse. Thus, instead of starting at 0, the counter will start at a number which is equal to 10 minus the inverse of the sample rate. Thus, the frequency of the output of the synchronous upcounter will be equal to the sample rate clock divided by the sample rate, or one MHz.

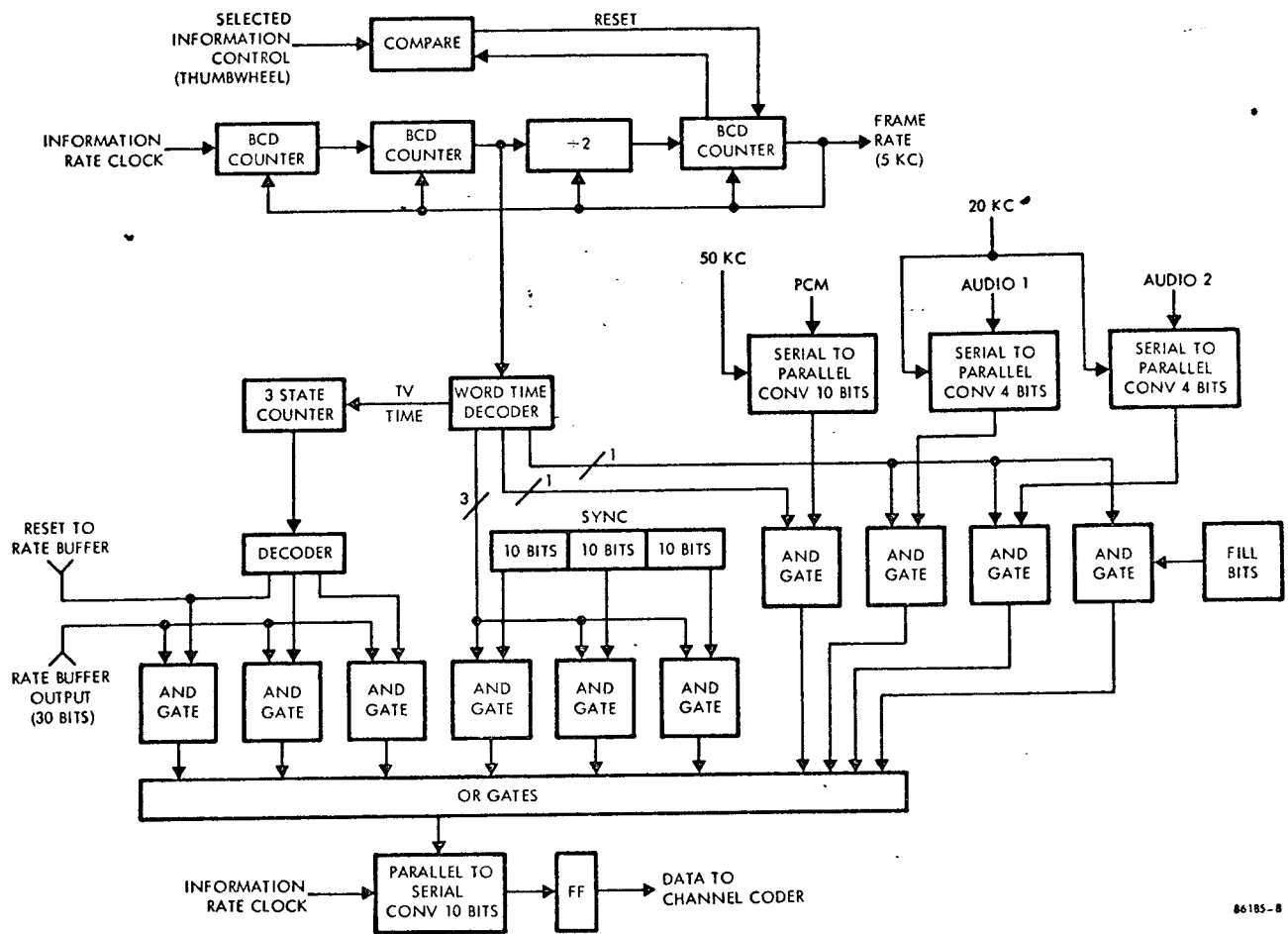
A2-2.12 Time Division Multiplexer - A3A2A1

The function of the Time Division Multiplexer is to generate a serial stream of continuous bits in the proper format and at the selected rate to the Channel Encoder.

The data from the audio sources, the PCM source and the TV data from the Rate Buffer are time division multiplexed in a predetermined format sequence. A predetermined sequence of 30 bits for frame synchronization is generated at the beginning of each frame in order for the demultiplexer in the receiver to identify the start of each frame. The multiplexer (Figure A2-2.12) is designed such that the frame rate remains constant for different selections of information bit rate. If, for example, the information bit rate is changed from 4 MHz to 8 MHz, the number of bits in the frame are doubled such that the frame rate does not change.

A2-2.12.1 Format

The format of the data frames is as follows (see Figure A2-2.12.1): The first 30 bits of the frame are devoted to the sync pattern. Bits 31 through 34 are for Audio 1 source.



86185-8

Figure A2-2.12. Time Division Multiplexer

1 SAMPLE = 10 BITS

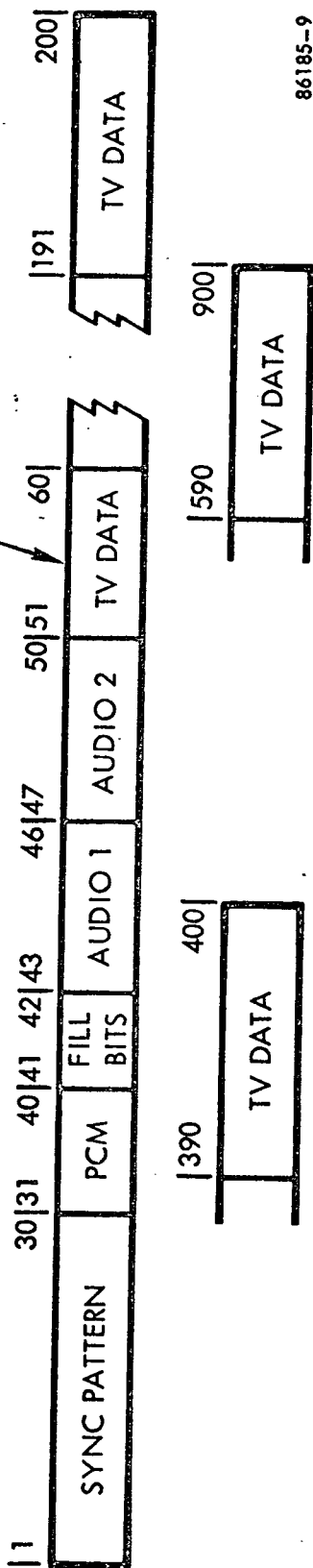


Figure A2-2.12.1. Time Division Multiplexer Format

Bits 35 through 38 for Audio 2 source. Bits 39 and 40 are fill bits. Bits 41 through 50 are for the PCM source. The remaining bits (150) are for TV data. Each data sample and associated time tag occupies 10 bits. Thus, an information bit rate of 1 MHz has a frame length of 200 bits. If the information bit rate was 2 MHz, the frame length would increase to 400 bits. The first 50 bits of the frame would remain the same as the 200 frame length case, but 200 bits of TV data would be added to the frame length. This procedure continues and for the case of 9 MHz selected information rate, the frame length would be 1800 bits. Thus, the frame rate remains constant at 5 kHz for any selected bit rate.

A2-2.12.2 Frame Rate Counter

The overall timing for the multiplexer is derived from the information rate clock and the selected information rate control. The information rate clock comes from the transmitter clock subsystem and covers a range from 1 MHz to 9 MHz in 1 MHz increments. The selected information rate control is four lines of BCD information from the front panel. Since the frame rate remains at a constant 5 kHz for any selected information rate, the selected control represents the multiple of 200 bits in a frame. The frame rate counter consists of 3 decade counters and a divide-by-2 counter. At the start of a frame the counters are reset and start from a count of zero. The counter is incremented by the information rate clock. The third decade counter is compared with the BCD number for the selected rate control and is allowed to increment until the count equals the BCD number from the selection rate control. Thus, the third decade counter is being reset at a rate equivalent to some multiple of 200 bits.

A2-2.12.3 Data Source

There are four sources of data which comprise the Time Division Multiplex format:

a. Sync Pattern

The sync pattern 3564622552 (octal) is inserted in each frame of the format. These sync bits are hard-wired to the inputs of the multiplexing gates.

b. Audio Channels

The multiplexer interfaces with two sources of 20 kHz digital audio data. A continuous 20 kHz clock is presented along with the audio data to the multiplexer. Since the frame rate is 5 kHz, four bits of audio data are generated from two channels during each frame. A four-bit serial-to-parallel converter is provided for each of the audio channels and the input data is shifted into these registers at the constant rate of 20 kHz.

c. PCM Data

The multiplexer interfaces with a 50 kHz serial PCM channel. A continuous 50 kHz clock rate is also presented from the multiplexer to the external unit. The 50 kHz clock shifts the PCM data into a 10-bit serial-to-parallel converter, composed of 2 5-bit shift registers.

d. Rate Buffer Output

The multiplexer interfaces with the 30 parallel data bits of the output of the rate buffer. The 30 bits are transferred into the multiplexer 10 parallel bits at a time. After the last 10 bits are transferred into the multiplexer, the multiplexer generates a reset pulse to the rate buffer which enables the rate buffer to access the next memory location and provides the next 30-bit word to the multiplexer.

A2-2.12.4 Multiplexing Operation

A three-stage counter provides a 0-7 count to the 10 MSI multiplexers. These multiplexers decode the count and individually enable the data inputs. The counter is incremented every 10 bits by the output of the first BCD counter in the Frame Counter. At the start of the frame (with the 3-stage counter at zero) the first 10 bits of the sync pattern are transferred into the output register. A bit count of 9 is detected, disabling the serial operation of the output register and enabling the parallel input of the next 10 bits of the sync word. The serial data flow is uninterrupted due to the output flip-flop which holds the tenth bit during the load time. Continuing in this manner, the third set of 10 bits from the sync pattern is generated, followed by the PCM and audio data.

After the first 50 bits have been multiplexed, TV data will be inserted into the bit stream in 10-bit groups for the balance of the frame. This occurs on counts 5, 6 and 7 of the 3-stage counter. The counter will recirculate on these three counts until the end of the frame, at which time the state of the counter is stored in two D flip-flops. The 3-stage counter will be reset and the frame counter will start at zero, stepping the 3-stage counter every 10 bits as before. However, after the first 50 bits have been generated into the bit stream the 3-stage counter will step to the count it was in at the end of the previous frame, thus preventing loss of continuity in the TV data.

A2-2.13 Convolution Encoder - A3A3A3

The functions of this unit are as follows:

- a. Accepts serial data and clock (to 9 MHz) from multiplexer.
- b. Accepts 1/2 rate or 1/3 rate clock from timing generator.

- c. Accepts Fano/Viterbi enable from control panel.
- d. With Fano enable and $1/2$ rate clock it generates systematic codes with constraint length up to 40 and polynomial terms up to 40 (switch selectable) at input data rates to 9 MHz (i.e., output to 18 MHz).
- e. With Viterbi enable and $1/2$ or $1/3$ rate clock it will generate nonsystematic codes with constraint length up to 6 and polynomial terms up to 6 (switch selectable).
- f. Independent polynomial programming is provided for Fano and Viterbi so that rapid change between Fano and Viterbi decoders can be made.

A2-2.13.1 Detail Description

The Convolution Encoder (Block Diagram, Figure A2-2.13.1) consists of a 40-bit shift register, one bank of 40 polynomial switches, three banks of 6 polynomial switches, one 40-bit modulo 2 adder, three 6-bit modulo 2 adders and an output selecting 4-bit shift register.

In operation, serial data is loaded at information bit rate into the 40-bit shift register. Each stage of this register drives 2 input gates which are enabled by the polynomial programming switches. These gate outputs are wired directly to the four independent modulo 2 adders. The modulo 2 adders outputs are identified as P_F , P_{V1} , P_{V2} , and P_{V3} .

When Fano has been selected, the data bit and P_F are parallel loaded into the output shift register at information data bit rate and are positioned to output the data bit first at rate $1/2$.

When Viterbi has been selected, P_{V1} , P_{V2} , and P_{V3} are parallel loaded into the output register at information data bit rate and are positioned for output in that sequence at rate $1/2$ or $1/3$. (Note: Only P_{V1} and P_{V2} get shifted to the output stage when rate $1/2$ is selected.)

Feedback connections are provided so that the 40-bit shift register can be operated as a length 63 pseudonoise (PN) sequence generator, by the addition of a jumper wire.

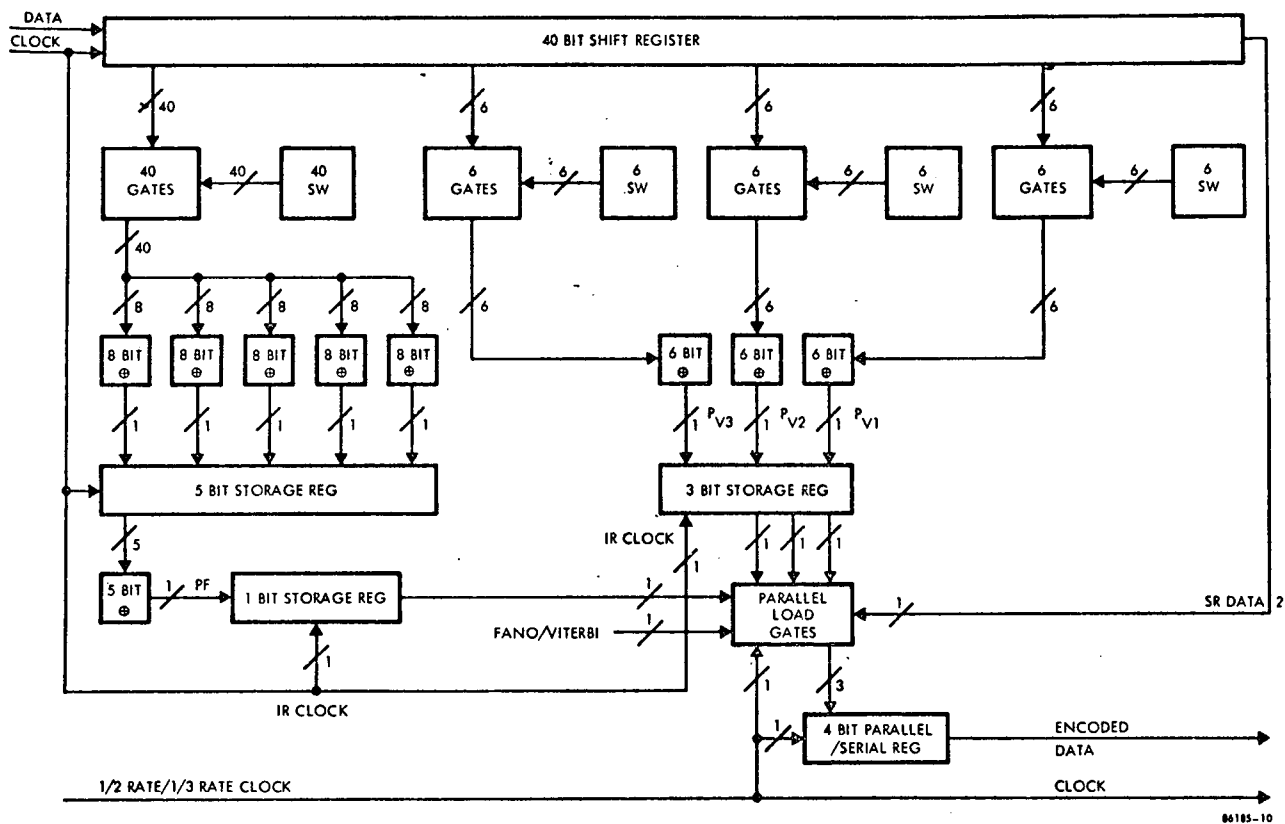


Figure A2-2.13.1. Block Diagram Convolution Encoder

A2-2.14 Noise Generator - A3A3A2

The noise generator is to provide a random noise signal with programmable amplitude which can be summed with the output signal in the Transmitter Output Buffer to simulate a noisy data link.

The noise generator consists of a noise source and a programmable attenuator. The noise source has a 25 MHz bandwidth and a 1 V rms output amplitude. The attenuator is variable from 0 dB to 22.1 dB in 0.1 dB increments.

A2-2.15 Output Buffer Amplifier - A3A3A1

The transmitter output buffer sums data from the Convolution Coder and noise from the Noise Generator to generate an output data stream consisting of data plus noise. The data amplitude and the output signal dc offset are controlled by front panel settings.

The transmitter output buffer (Drawing 528732) consists of a wideband operational amplifier with a gain control on the data input and a fixed gain on the noise input. The gain of the data input is controlled by a "Raysistor" in parallel with a fixed resistor. The impedance of the Raysistor varies inversely with the voltage across its input. This input voltage is controlled by an op amp whose voltage is set by a front panel control. The effect of varying the front panel gain pot is to vary the voltage across the Raysistor and so to reduce its impedance. When the impedance of the Raysistor is low, the gain of the output buffer amplifier from the data input to the data output is high. When the voltage across the Raysistor is zero, as it would be when the front panel pot is set to zero, the gain of the data input to the data output is fixed by the ratio of the input resistor to the feedback resistor.

The output offset is controlled by another front panel pot which sums a current into the summing junction of the wideband op amp through a resistor. The offset may be varied from approximately +4 V to -4 V at the data output by means of this front panel control.

A2-3.0 DETAIL RECEIVER DESCRIPTION

The following sections describe the individual units of the Receiver.

A2-3.1 Bit Sync-Signal Conditioner - Preprocessor - A1(A2)

In Figure A2-3.1 the block diagram for a High Bit Rate signal conditioner, bit sync, and preprocessor is given. The signal conditioner consists of the matched filter and the multilevel decision unit. In the multilevel decision unit seven threshold detectors are used in conjunction with coding logic to effect an eight-level or three-bit decision on the state of the matched filter output at the end of a bit period. Timing is controlled by the clock coming from the bit sync.

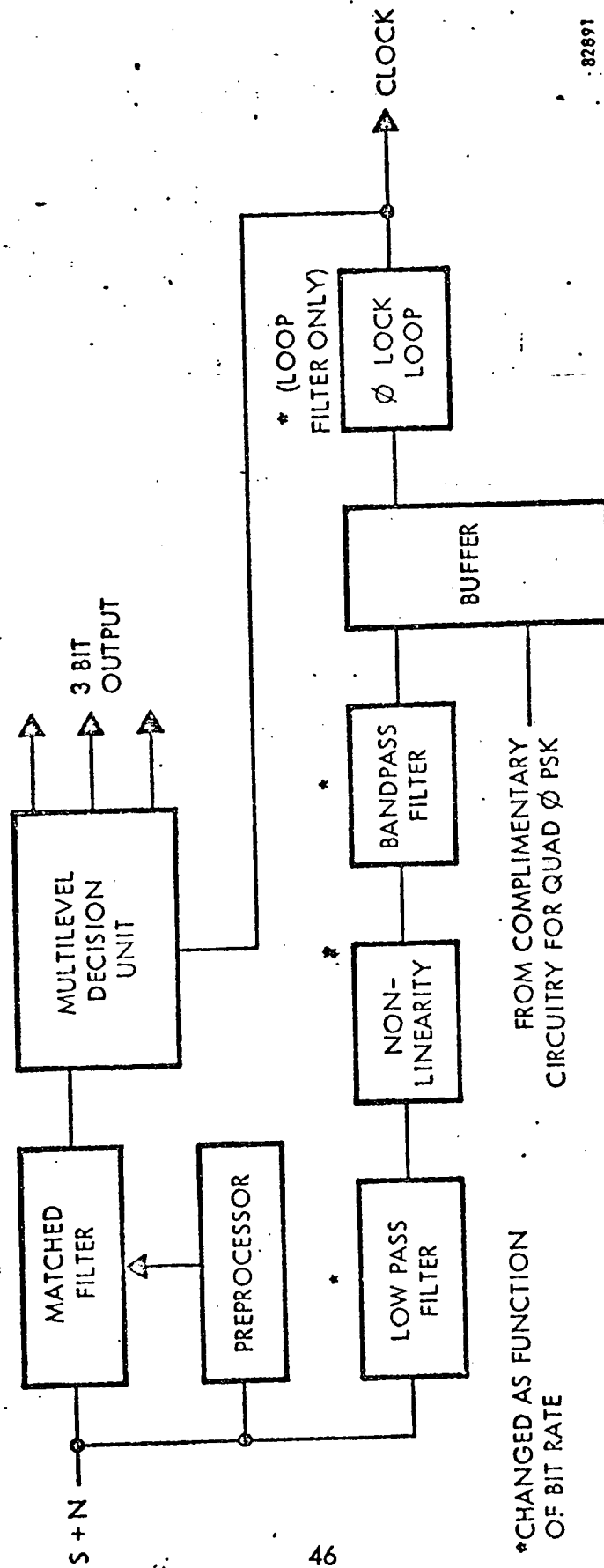
In this application, the only preprocessing required is baseline correction, (DC Restoration). In the preprocessor the positive and negative peaks of the noisy PCM are measured and used to effect baseline correction.

For bit synchronization the matched filter output is passed through a circuit which generates a bit rate spectral component. This signal passes through a bit rate center frequency bandpass filter. The output of the filter is the input to a phase-lock loop where a vco is locked to the bit rate component.

Those items with an asterisk in the block diagram are the functions which must be changed in order to change bit rates. These functions are all contained in a bit rate plug-in module.

For quadrature operation a second BSSC system is employed. The data from the two units are combined in the Quad Phase Data Combiner, described in Paragraph A2-3.2.

The BSSC system can also be operated at a bit rate other than that of the plug-in employed by use of an external clock. A circuit is provided allowing the operator to select the bit synchronizer lock or the external clock. Thus, in this mode only the bit detector or signal conditioner is used. Since the matched filter is designed to maximize signal-to-noise ratio at a specific bit rate, some performance degradation will result when the filter is used at another bit rate. The predicted degradation for the 27 MB and 18 MB plug-ins for various bit rates is given in the following.



82891

Figure A2-3.1. Signal Conditioner, Bit Sync, Preprocessor

27 MBit Plug-In Installed

Trans. Bit Rate (MHz)	Degradation (dB)
27	0
24	0.52
21	1.09
18	1.76
16	2.27
15	2.56
14	2.86
12	3.52
10	4.32
9	4.76
8	5.27
6	6.52
4	8.27
3	9.52
2	11.27

18 MHz Plug-In Installed

Trans. Bit Rate (MHz)	Degradation (dB)
18	0
16	.51
15	0.8
14	1.1
12	1.76
10	2.56
9	3.0
8	3.51
6	4.76
4	6.51
3	7.76
2	9.51

More details of the Bit Sync/Signal Conditioner are contained in the Operating and Maintenance Manual for this unit.

A2-3.2 Quadrphase Data Combiner - A9A2

The function of this unit is to interface between the two parallel Bit Synchronizers utilized in the quadrphase mode, and the serial input to the convolution decoders. To accomplish this, the following operations must be performed:

- a. The clock from Bit Sync A must be frequency doubled to generate a serial clock for the convolution decoders.
- b. The data from the two Bit Synchronizers must be combined or multiplexed into a single serial (3 bits wide) data stream.
- c. Provision must be to correct for all possible states of demodulator lock-up.

A block diagram of the unit is shown in Figure A2-3.2. Clock from Bit Sync A is frequency doubled in the Clock Rate Multiplier and routed to internal registers and the Clock Selector. Data from the two Bit Syncs is buffered in Input Registers, combined in the Multiplexer, and stored in the Intermediate Data Register. The output of this Register, together with the output of the Data A Input Register is applied to the Output Selector. This Selector is controlled by the Biphase/Quad Phase switch, which also selects between Bit Sync A clock, or the frequency doubled clock.

The required operations for the ambiguity resolver can be understood from the following table. Assume that data streams I and Q are transmitted. These may appear at Bit Sync A and B outputs in any of the following forms, depending on the state in which the Demodulator locks.

<u>Case</u>	<u>Bit Sync A</u>	<u>Bit Sync B</u>
1	I	Q
2	Q	\bar{I}
3	\bar{I}	\bar{Q}
4	\bar{Q}	I

Each data stream may appear at either Bit Sync, either inverted or noninverted. Case 3 is the inverse of case 1, and case 4 is the inverse of state 2. These complete inversions are handled after the data is multiplexed in the Decoders. Thus, only the switching between sync cases 1 and 2, the order reassignment and inversion of one channel, is performed in the Combiner. This switching is performed in response to signals generated by the active convolution Decoder and processed by the Quad Phase Ambiguity Correction Flip-Flop.

A2-3.3 Viterbi (Maximum Likelihood) Decoder

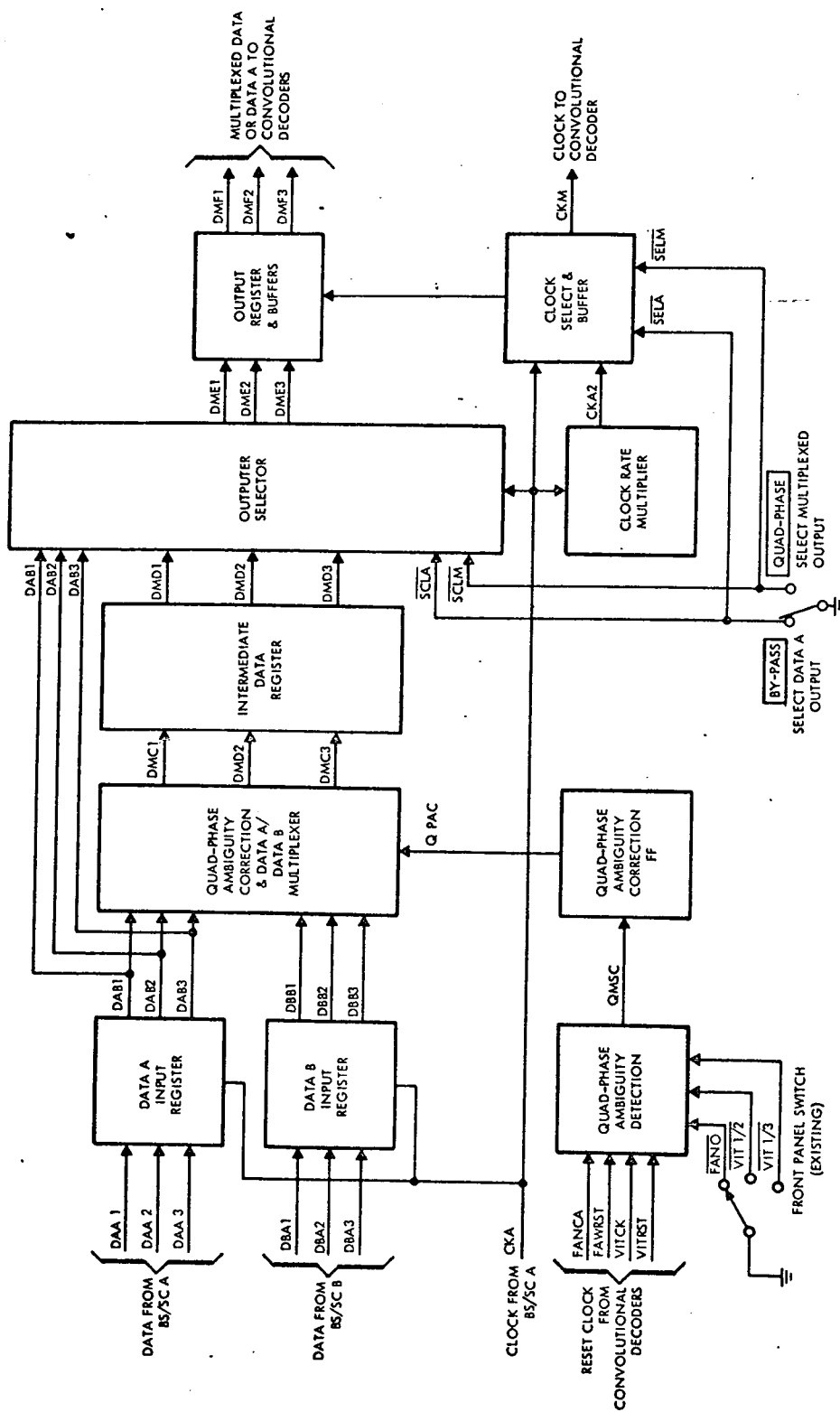
This section reviews briefly some aspects of convolution codes and maximum likelihood decoding, discusses the general functional organization of Viterbi Decoders, and then describes the specific hardware in the Test System.

A2-3.3.1 Review of Decoding Process

The encoder for a rate $1/m$ convolutional code of constraint length k can be modeled as a finite state machine possessing 2^{k-1} states. Transitions from a given state are always into one of two possible states and transitions into a given state are always from one of two possible states. The coder input determines which transition occurs and the transition uniquely determines the m -bit coder output. In Figure A2-3.3.1-1 we show a rate $1/2$, $k = 3$ encoder and the associated finite state machine in Figure A2-3.3.1-2. Figure A2-3.3.1-3 is called a trellis diagram and is an alternate representation of the finite state machine in which a new set of states have been drawn for each possible time increment. The operation of the maximum likelihood decoder is probably most easily understood in terms of the trellis diagram and proceeds in the following manner.

Starting from a known state (normally the all zero state at the beginning of the message) the decoder hypothesizes the two possible extensions and computes and stores a metric proportional to the log of the probability that the actual received m -bit branch was received given the hypothesized branch was transmitted. It then proceeds to the next received branch, extends each of the two hypothesized branches along the two possible paths, computes the appropriate branch metrics and adds these to the previously stored values to produce 4 path metrics. The decoder continues in this fashion until it has hypothesized 2^{k-1} paths and has computed 2^{k-1} path metrics. Thereafter, the decoder extends each hypothesized sequence along the two possible paths, computes the new path metrics, compares the two path metrics incident at each of the 2^{k-1} states and for each state discards the path having the smallest metric. Thus, the decoder is constantly maintaining 2^{k-1} hypothesized sequences and 2^{k-1} path metrics. At each step in the decoding procedure the paths which are discarded are those paths which cannot possibly end up being the maximum likelihood choice since all possible extensions of the retaining paths will always possess larger path metrics.

In Viterbi's original paper the actual sequence was selected by terminating the code word with a known set of $k-1$ information bits. Consequently, when each of the last $k-1$ branches was processed at the decoder, the paths were only extended along the branch known to have been sent and after $k-1$ iterations the set of 2^{k-1} sequences was reduced to one sequence. This procedure, although academically satisfying, was unhandy since it required one to maintain code word synchronization and also resulted in code rates that were not the ratio of small whole numbers thus necessitating rather complicated rate conversions. It was subsequently recognized that it was unnecessary to terminate the code and that equally good results could be obtained with unterminated codes and a fixed decoding delay L on the order of 5 or 6 constraint lengths. The newer procedure was to maintain only the most recent L bits of each hypothesized sequence. After each new branch was received and the sequences updated, the sequence possessing the maximum path metric was selected and the last bit in this sequence was outputted while the last bit in each



98185-11

Figure A2-3.2. Block Diagram Quad-Phase Data Combiner

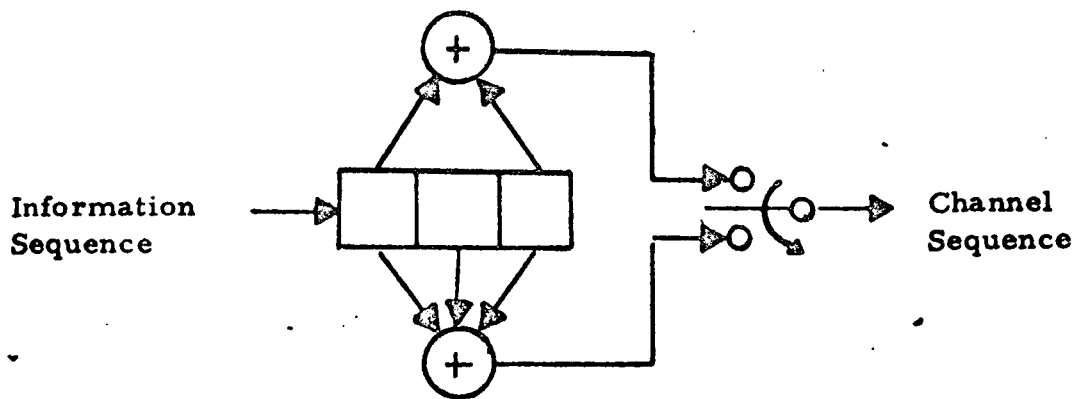


Figure A2-3.3.1-1. $k = 3$, Rate $1/2$ Convolutional Code

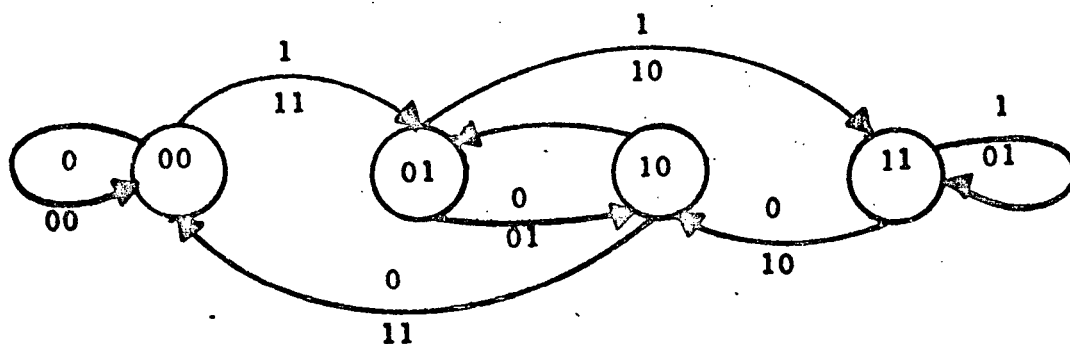


Figure A2-3.3.1-2. State Diagram for $k = 3$, Rate $1/2$ Convolutional Code

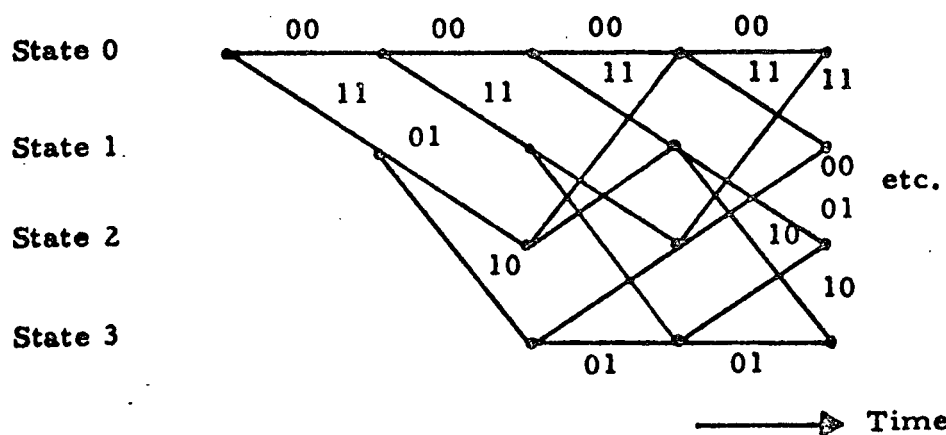


Figure A2-3.3.1-3. Trellis Diagram for $k = 3$, Rate $1/2$ Convolutional Code

of the other sequences was discarded. This procedure obviously causes a slight performance degradation in that it only approaches the maximum likelihood case as $L \rightarrow \infty$. Finite storage is, however, a physical necessity and the procedure has the very definite advantage that it avoids the rate loss associated with periodic resynchronization as well as the problem of rate buffering the code blocks and maintaining the required block synchronization. There are, however, cases in which the use of terminated codes is advantageous particularly in short burst communication systems and certain TDMA applications. Even in these cases, however, one would almost always decode the first portion of the received word with a constant decoding delay and only make use of the code termination to decode the last few constraint lengths of data.

A2-3.3.2 General Functional Organization of the Decoder

A machine which implements the decoding process described in the previous section needs to perform the following functions.

It accepts at the input a continuous sequence of n -bit binary numbers which represent the successive outputs of the channel bit detector quantized to one of 2^n levels and delivers the decoded information stream at the output. The machine itself consists of (1) a synchronizer which provides both branch timing information to the decoder and determines the correct bit polarity if this is required, (2) a branch metric computer which determines the appropriate branch metrics for each received code branch, (3) a path metric updating, comparison and storage device, in which the branch metrics are added to the previously stored path metrics, the appropriate comparisons made and the new path metrics stored, (4) a device for updating and storing the hypothesized information sequences, and (5) an output decision device which provides the decoder output and also provides information to the synchronizer. Each of these devices is discussed in turn in the remainder of this section and the design tradeoffs are elaborated upon.

A2-3.3.2.1 Decoder Input

The decoder input is provided in the form of an n -bit binary number by the bit detector. The bit detector would normally consist of a matched filter (or suitable approximation) followed by an A/D converter.

There are three interrelated parameters which need to be determined. These are the number of quantization levels, the spacing between quantization levels, and the number of bits that will be used to represent the branch metric. A particularly simple ad hoc scheme is to use 3 bits (8-level quantization) with equally spaced levels, and a uniform 3-bit metric associated with a single channel bit. That is, the metric would be allowed to take on the values 0 through 7 depending on how closely the received channel bit matched with the hypothesized channel bit. Computer simulations have verified that with additive Gaussian noise this scheme works very well and provides a level of performance that differs from that possible with extremely fine quantization by only a few tenths of a dB. Computer simulations have also verified that the quantizing range is not critical and that the probability of decoded bit error exhibits a broad minimum as a function

of the spacing between levels with a near optimum setting occurring when the highest slicing level in the A/D converter is chosen to be $1/2$ a quantization level below the expected value of the received signal.

A2-3.3.2.2 Branch Synchronizer

The synchronizer is simply a device for inserting commas (determining the beginning and ending of a branch) in the received bit stream and resolving bit polarity when required. The operation of the synchronizer is straightforward and simply requires an indication that the input sequence contains a much larger number of errors than normal. Several techniques are possible and almost any way that one can get an error detecting indication will work. One method for which there are many possible variations is to look at the rate at which the path metrics are increasing. A second method and one that is particularly simple to implement is to check the oldest bit in each of the 2^{k-1} hypothesized bit streams and determine whether or not all of the bits are unanimous. In the absence of excessive channel noise, the number of nonunanimous decisions is very small whereas with a bit slip or polarity inversion the effective channel noise is sufficiently large that nonunanimous decisions will occur a large percentage of the time. Consequently, one need only count the number of unanimous decisions that occur in some fixed block length and compare the value with a threshold.

A2-3.3.2.3 Branch Metric Computer

For rate $1/m$ codes, 2^m different branch metrics need to be computed each bit time. If the uniform assignment method discussed previously is used, then the computation is relatively straightforward. For example, if the convention is adopted that the binary number 000 corresponds to a highly reliable received "0" and the numbers 001, 010, etc., indicate received zeroes with decreasing reliability, then the branch metric computer simply inverts this number to obtain the channel-bit metric associated with a hypothesized "0" or does nothing to it to obtain the channel-bit metric associated with a hypothesized "1." To compute the branch metric, it need only sum the individual bit metrics that make up the branch.

A2-3.3.2.4 Path Metric Updating and Storage

In this portion of the system the received branch metrics are added to the previously accumulated path metrics, the appropriate path metrics are compared, and the largest value in each comparison is retained. Both this portion of the machine and the portion that stores the hypothesized information sequences account for the greatest portion of the parts that are used so it is in these two areas that the greatest care need be taken in design in order to avoid excessively large part counts. A useful modularizing concept can be developed if we number the nodes in the trellis diagram from 0 to $2^{k-1} - 1$ starting with the uppermost node. In this case it is easily verified that the two descendants of states j and $j + 2^{k-2}$ ($0 \leq j < 2^{k-2}$) will always be states $2j$ and $2j + 1$ where state $2j$ is always the "0" extension of j and $j + 2^{k-2}$ and $2j + 1$ is the "1" extension. Thus, one may implement this portion of the decoder by repeated use of a module

which enables one to determine the four possible path metrics corresponding to extensions of states j and $j + 2^{k-2}$, make the required comparison and restore the new path metrics for states $2j$ and $2j + 1$. The repeated use of the module may be accomplished by time sharing for a very low speed machine or by actual duplication of the module for very high speed. In between a large number of serial and parallel combinations are possible and a detailed discussion of the tradeoffs is beyond the scope of this section. In addition, the basic module itself may be implemented using all parallel adds and compare operations or it may be done using a serial add followed by a serial compare. A still different variation is to implement the module so that the new metric for state $2j$ and $2j + 1$ are formed sequentially using the same module. In the actual machine discussed in the following section, we went somewhat beyond the "fully parallel" case and actually doubled up on this module so that the add and compare operations could be done simultaneously to achieve high speed operation.

The metric values themselves may be stored in a long serial register for very low speed operation, or in some combination of serial and parallel storage at higher speeds. Here, again, the number of combinations is very large and the optimum choice will depend on the specific application.

One further point needs elaboration, however, and that regards the size of the metric registers. Fortunately, the difference between the smallest and largest metric can easily be shown to be bounded by $(k-1)M$ where M is the maximum value of the branch metric. For a rate $1/2$, $k = 6$ code using a 3-bit metric for each channel bit, the value of M is 14 and thus the maximum difference is 70. Consequently, a 7-bit metric register is completely adequate although for convenience and ease of rescaling one would probably use 8-bit registers.

A2-3.3.2.5 Storage and Updating of Hypothesized Information Sequences

The basic modular aspect of this portion of the decoder is identical to that of the metric storage device discussed in the previous paragraph and in some cases the devices are combined. In concept this device consists of 2^{k-1} registers capable of storing L bits each. Normally one would reserve fixed storage locations for the sequence associated with each state although this is not necessary. The registers must be interconnected such that the contents of the registers corresponding to states j and $j + 2^{k-2}$ can be transferred to the registers corresponding to $2j$ and $2j + 1$. Control must be such as to allow for all four possible ways of transferring the sequences. In addition, a means must be provided for removing the oldest bit from each sequence during the transfer operation and appending a new bit. The new bit is always the binary number corresponding to whether the new sequence came from state j or state $j + 2^{k-2}$ and will be zero if the former is true and one if the latter is true. This bit does not correspond to the most recent bit in the trellis but to the one $k-1$ branches back.

A2-3.3.2.6 Output Device

Several techniques are available for providing the decoder output. The best, of course, is to select the sequence possessing the largest path metric and choosing the oldest bit in this sequence. In a high speed fully parallel machine, however, this requires a considerable amount of logic and may not be cost effective. In general, we expect all of the sequences will

have "merged" by the time the decoding depth has been reached so that all paths in the trellis collapse into a single path. In this case one could just as well choose an arbitrary sequence and output the oldest bit. A method somewhat inbetween the two from a performance viewpoint is to make a majority decision over all sequences. Many other combinations are obviously possible but probably not practical. Obviously, some degradation takes place by using one of the sub-optimum techniques, however, they may be overcome by extending the decoding length. If, for example, a decoding depth of say 18 is adequate to achieve a given probability of error using the maximum correlated sequence technique, then something on the order of 25 will be required using the "majority vote" technique and 30 using the arbitrary sequence technique. All manner of trades are available and the right choice depends upon the logic line and implementation rate.

A2-3.3.3 Block Diagram Description

Figure A2-3.3.3-1 shows a detail block of the Viterbi Algorithm Decoder. Data from the bit synchronizer via the Q-P Data Combiner is loaded into the branch synchronizer which establishes branch sync. The branch synchronizer output is stored in input registers. Eight 3-bit adders then calculate the incremental correlations. A group of registers called the SC registers store the survivor path correlations. At each branch the survivor path correlations for 2^k paths are calculated. These are compared and 2^{k-1} paths are discarded. This is performed by the comparators and SC selectors. The outputs of the comparators are fed to the register exchange logic (survivor sequence registers) where 2^{k-1} survivor paths are stored. The decoder output is determined by the most ancient bit stored in each survivor sequence. The output is determined by the logic state of the bit which is in the majority.

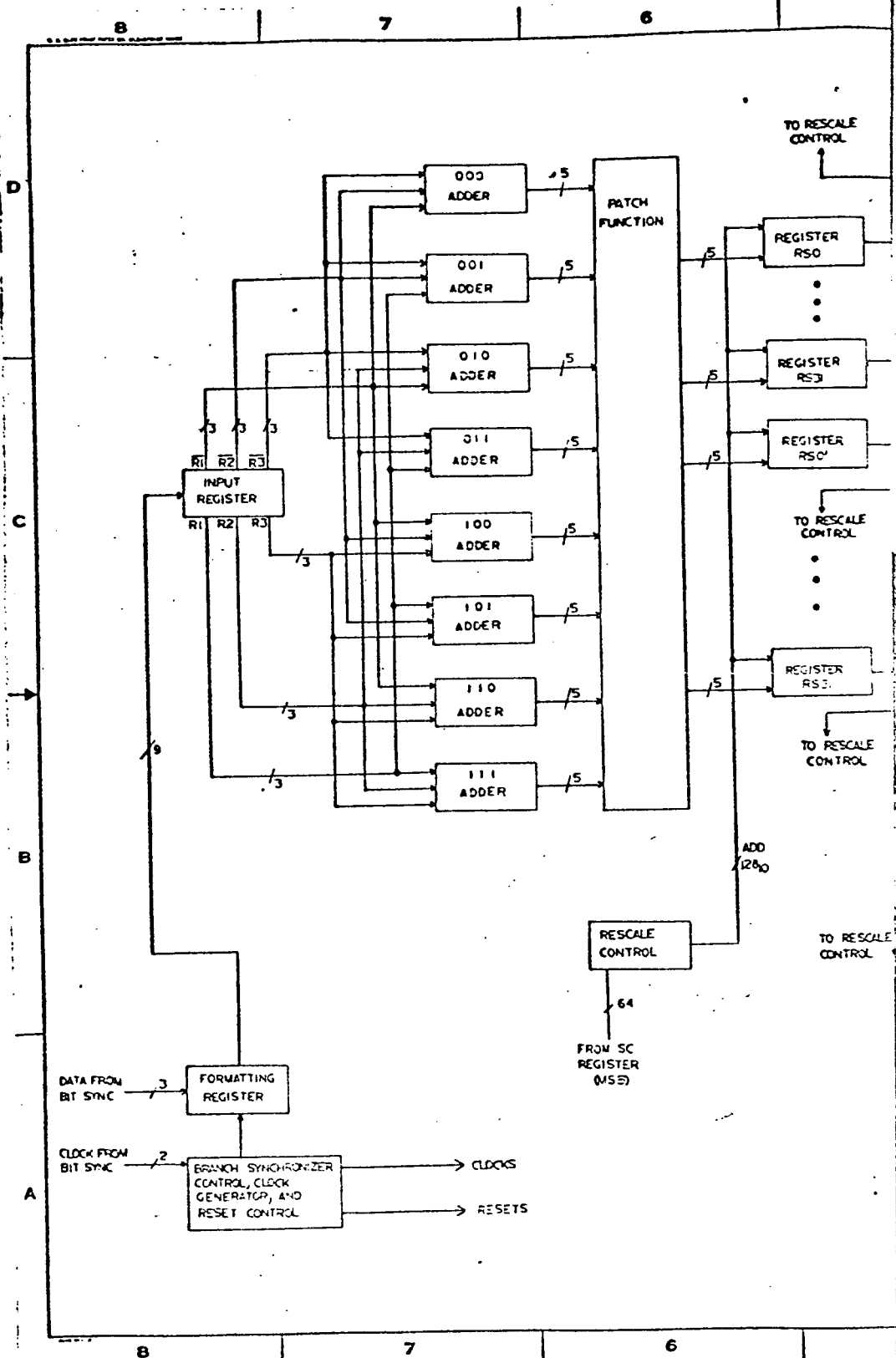
This decoder is designed to handle codes up to constraint length, $k = 6$, with three-bit quantization. Shorter constraint length codes are handled as special cases of $k = 6$ codes. Additional details will be included in the detail discussion of the various sections of the decoder.

The discussion to follow will be for the implementation of a 3-bit quantized, rate $1/3$, $k = 6$ decoder.

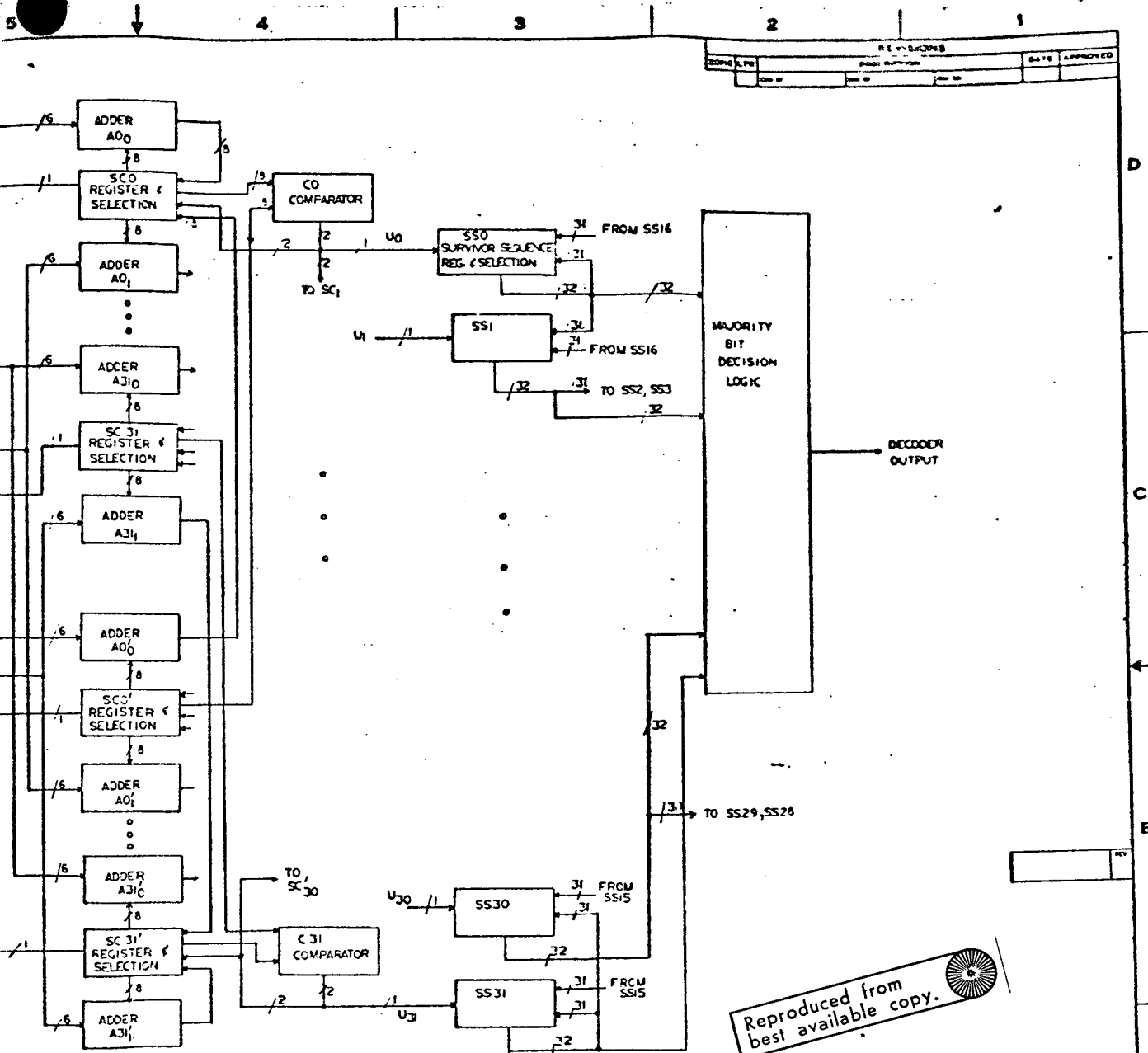
Reference may be made to the Viterbi Typical Timing Diagram (Figure A2-3.3.3-2).



FOLDOUT FRAME 1



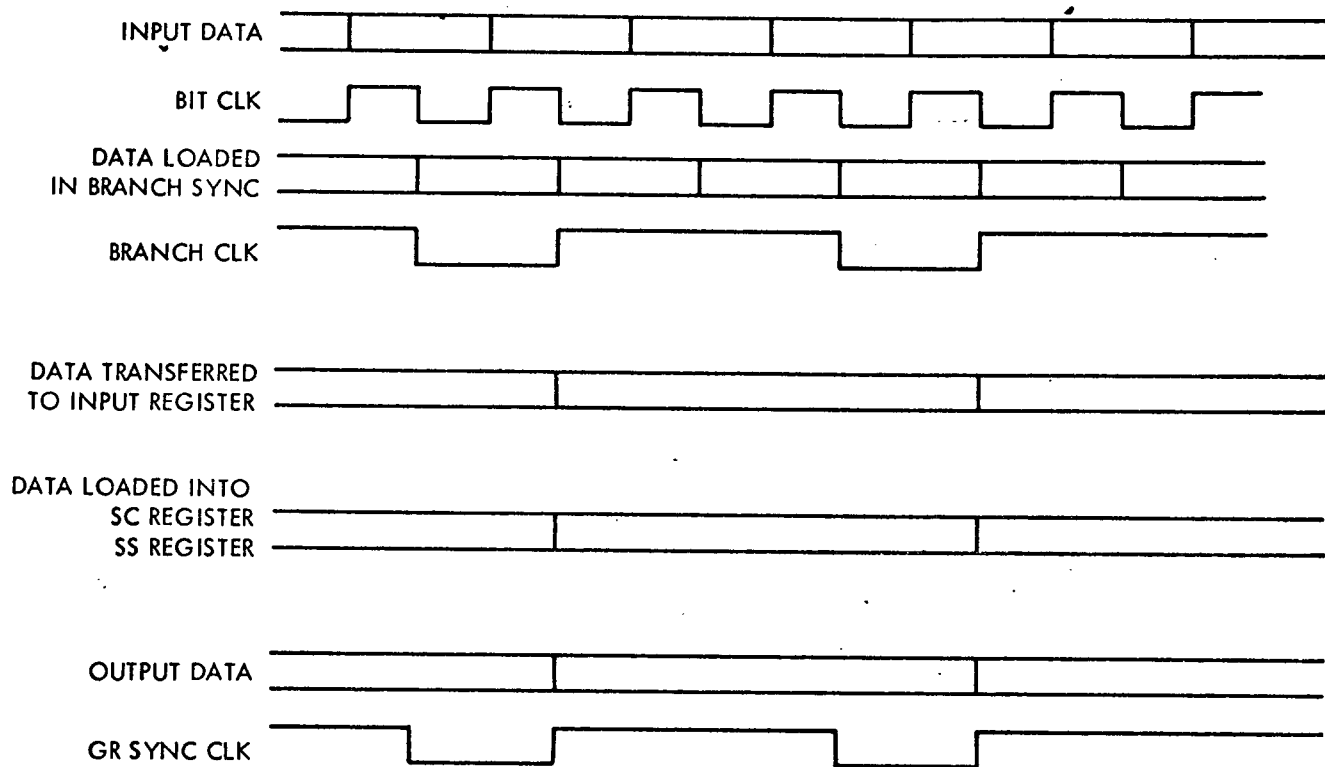
FOLDOUT FRAME 2



Reproduced from
best available copy.

QUANTITY REQUIRED		ITEM NO.		PART OR IDENTIFYING NO.		NOMENCLATURE OR DESCRIPTION		CODE IDENT	
LIST OF MATERIALS OR PARTS LIST									
PART TO BE FILED IN BLUE AND SHOWN IN RED DIMENSIONS ARE IN INCHES TOLERANCES ARE AS SHOWN UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE AS SHOWN UNLESS OTHERWISE SPECIFIED				DRAWING NO. 1577- SCALE 1:1 SHEET 1 OF 1					
DRAWING NO. 1577- SCALE 1:1 SHEET 1 OF 1				TITLE VITERBI DETAILED BLOCK DIAGRAM					
DRAWING NO. 1577- SCALE 1:1 SHEET 1 OF 1				SHEET CODE IDENT NO. 91417					
DRAWING NO. 1577- SCALE 1:1 SHEET 1 OF 1				SCALE 1:1					

PAGES 58 THROUGH 80 INTENTIONALLY LEFT BLANK



86185-12

Figure A2-3.3.3-2. Viterbi Typical Timing Diagram

A2-3.3.3.1 Branch Synchronizer

The decoder receives data from the bit synchronizer at bit rate and stores the data in the formatting register. The complete specification of a branch input consists of three 3-bit words for rate $1/3$. Each word is a 3-bit quantization of a bit in the received sequence. If branch synchronization has been established, the three words are associated with a branch in the code tree.

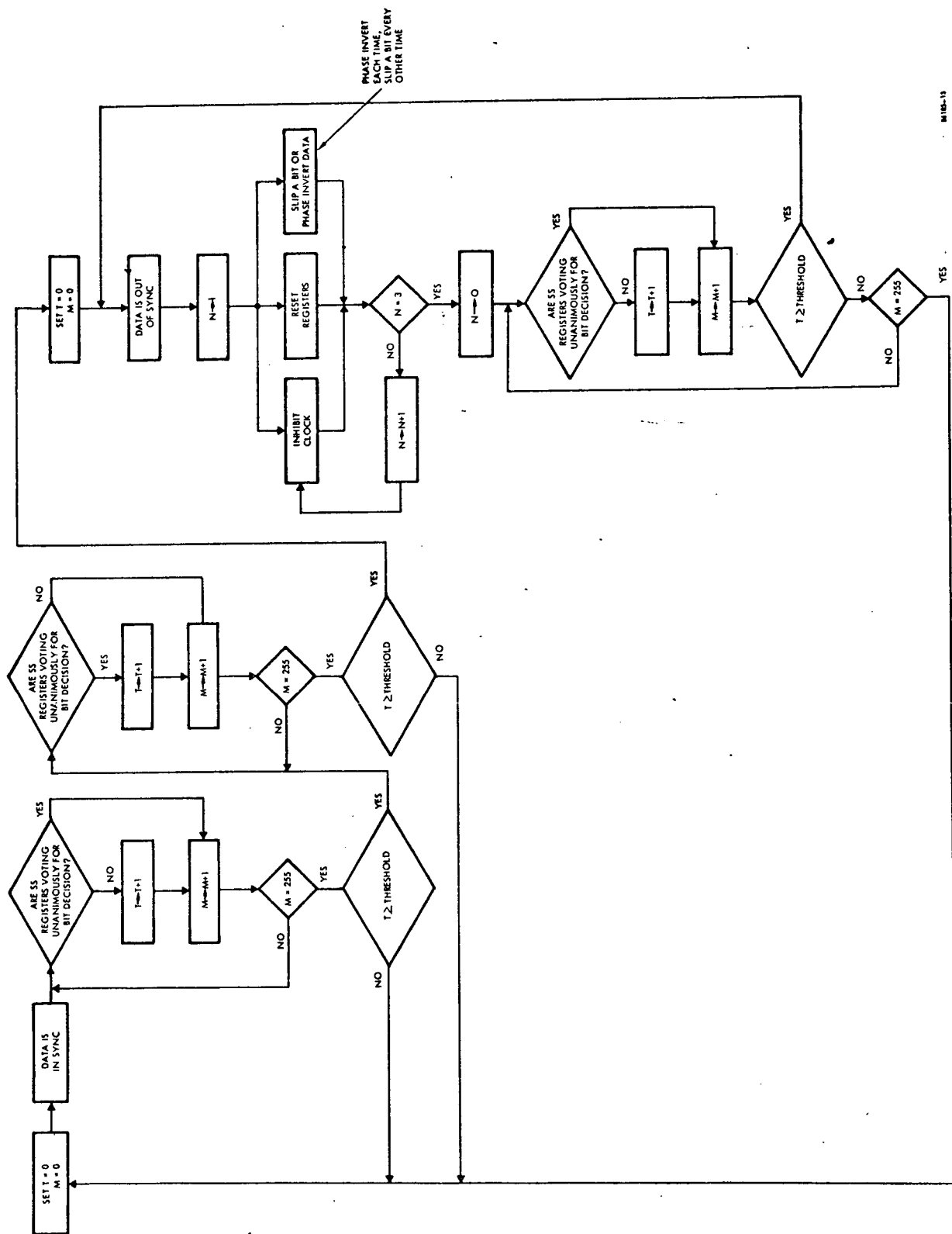
The quantized data is represented by a sign and two magnitude bits. A maximum one is 111 and a maximum zero is 011, a minimum one is 100 and a minimum zero is 000. However, the Viterbi operates using a code ranging from a maximum zero of 000 to a maximum one of 111. This conversion occurs in the formatting register. The formatting register is a temporary storage register that receives three 3-bit words and then transfers the 9 bits in parallel to the input register. This data has five ways to be out of synchronization: 1) phase inversion, 2) one-bit slip, 3) one-bit slip with phase inversion, 4) two-bit slip, 5) two-bit slip with phase inversion. For a rate $1/2$ code the two-bit slip modes are not possible.

The branch synchronizing is accomplished by a trial and error process. With high probability bits at various depths in the survivor sequences should be unanimous on the bit decision if branch synchronization has been established. These depths are 16, 22, 27, and 32 for $k = 3, 4, 5$, and 6, respectively. These depths are switch selectable by switches located on the logic panel.

The actual implementation uses two counters, one counts branch times and the other counts nonunanimous bit decisions. The branch synchronizer has two modes of operation: 1) search and 2) lock. When the branch synchronizer is in the search mode, it requires the threshold to be exceeded by the nonunanimous bit counter before trying another branch position. The branch synchronizer alternately phase inverts and slips a bit until sync has been established. The thresholds are 60 and 30 for $k = 5$ or 6 and $k = 3$ or 4, respectively. If the threshold is not exceeded during 255-bit times, then the branch synchronizer is assumed to be in the lock mode. Once sync has been established, the threshold must be exceeded twice before the branch synchronizer will try a new branch position (refer to branch synchronizer flow diagram Figure A2-3.3.3.1-1).

When the branch synchronizer goes to a new sync position, it generates a clock inhibit for three branch times and it also generates a reset which clears all registers. The inhibit allows the Viterbi time to update the incremental correlations before processing resumes. The branch synchronizer slips a bit by generating a blanking pulse which inhibits the input clock to the clock generator for one-bit time. Reference may be made to Branch Synchronizer Typical Timing Diagram (Figure A2-3.3.3.1-2).

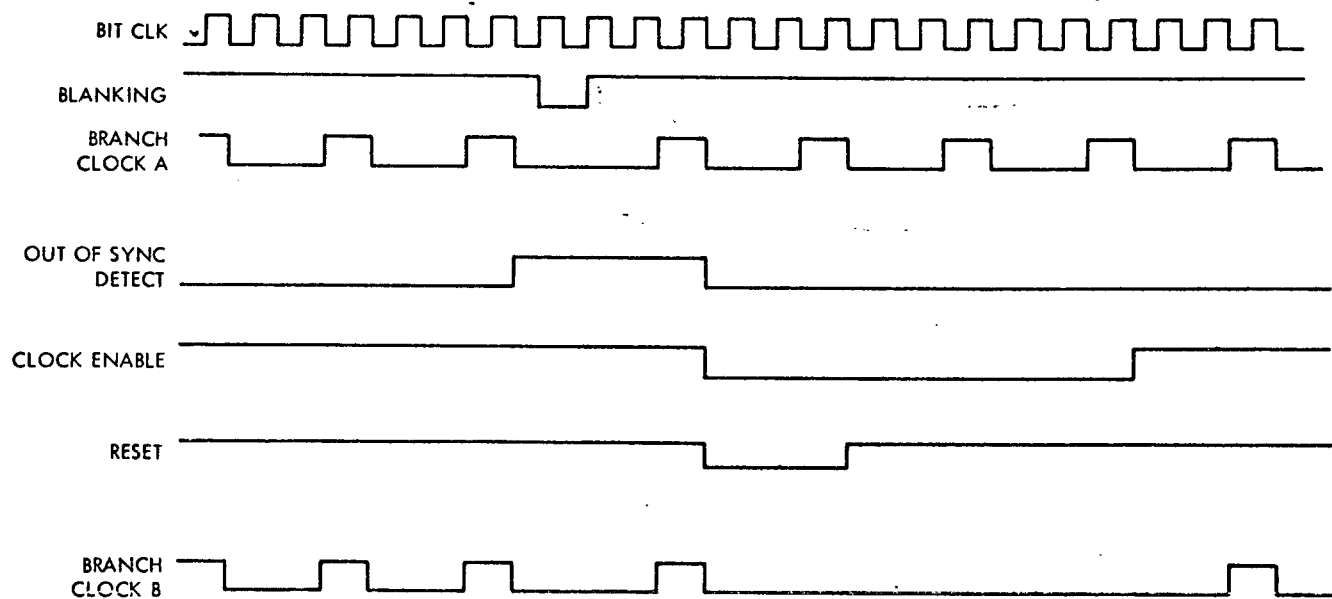
For rate $1/3$ codes, if all odd-weight generators are used, the Viterbi cannot distinguish between the in sync mode and the mode with no bit slippage with phase inversion. That is, the Viterbi could output either true or complement data under these conditions.



M = BRANCH TIME COUNTER
N = BRANCH TIME COUNTER
T = UNANIMOUS DECISION COUNTER

M105-13

Figure A2-3.3.1-1. Branch Synchronizer Flow Diagram



86185-14

Figure A2-3.3.3.1-2. Branch Synchronizer Typical Timing Diagram

A2-3.3.3.2 Clock Generator

The clock generator receives the bit sync clock and divides its frequency by 3 for rate 1/3 or by 2 for rate 1/2. The clock generator also provides the necessary drive for the clock lines. The code rate is selectable by a switch on the front control panel.

A2-3.3.3.3 Input Register

The input register consists of three 3-bit registers containing the branch data. Phase inversion for branch synchronization is performed at the inputs to the input register. Both true and complement data are provided to the incremental correlation calculator. If a rate 1/2 code or less than 3 levels of quantization is used, then bits of the register are not used. Both true and complement outputs of these unused bits are forced to a logical "1" so that they add equally to all correlations. The selection of the quantization is controlled by two switches located in the logic.

A2-3.3.3.4 Calculation of Incremental Correlations

The first rank of eight adders shown in Figure A2-3.3.3-1 are labeled 000, 001, ..., 111. These adders compute the approximate correlation between the received 3-bit branch and the 8 possible branches (rate 1/3). The adder labeled 000 calculates the correlation between the received 3-bit branch and the 000 branch. Similarly, the other adders are labeled by the branch with which the received branch is being correlated.

The quantized outputs are represented by 3-bit numbers ranging from 000 to 111. The 000 sample corresponds to the most negative quantized sample and 111 corresponds to the most positive quantized sample.

The adder labeled 111 calculates the correlation between the received branch and the 111 branch. Note that the inputs to the adders are r_1 , r_2 , and r_3 . The values of r_1 , r_2 , and r_3 range from 000 to 111. If the received branch was 111, in the noise-free case all quantized outputs would be their maximum positive value (111). Then the binary sum of $r_1 + r_2 + r_3 = 10101$ or 21_{10} . If noise is present on the channel, the samples r_1 , r_2 , r_3 may not be their full positive value, but would be smaller quantities ranging from 000 to 111. In the noisy case the correlation may be less than 21_{10} .

The adder labeled 000 calculates the correlation between the received branch and the 000 branch. Note that the inputs to the adder are \bar{r}_1 , \bar{r}_2 , and \bar{r}_3 , the one's complements of r_1 , r_2 , and r_3 . The values of \bar{r}_1 , \bar{r}_2 and \bar{r}_3 range from 000 to 111. If the received branch was 000, the noise-free case all quantized outputs would be their maximum negative value (000). Then the binary sum of $\bar{r}_1 + \bar{r}_2 + \bar{r}_3 = 10101 = 21_{10}$. If noise is present on the channel, the samples r_1 , r_2 , r_3 may not be their full negative value, but would be smaller quantities ranging from 000 to 111. In the noisy case the correlation may be less than 21_{10} .

The other adders are labeled by the branch with which the received sequence is to be correlated. Note that the inputs are arranged to accomplish the proper correlation.

The adders outputs are called incremental correlations. The outputs are binary numbers between 0 and 2^{10} .

The addition is performed in two branch times because of speed considerations. The sum of the three numbers is performed by first adding r_1 and r_2 , then storing the 4-bit partial sum and r_3 . In the next branch time, the 4-bit sum and r_3 are added and stored. The two branch times of delay do not cause difficulties in further processing.

A2-3.3.3.5 Rescale Control

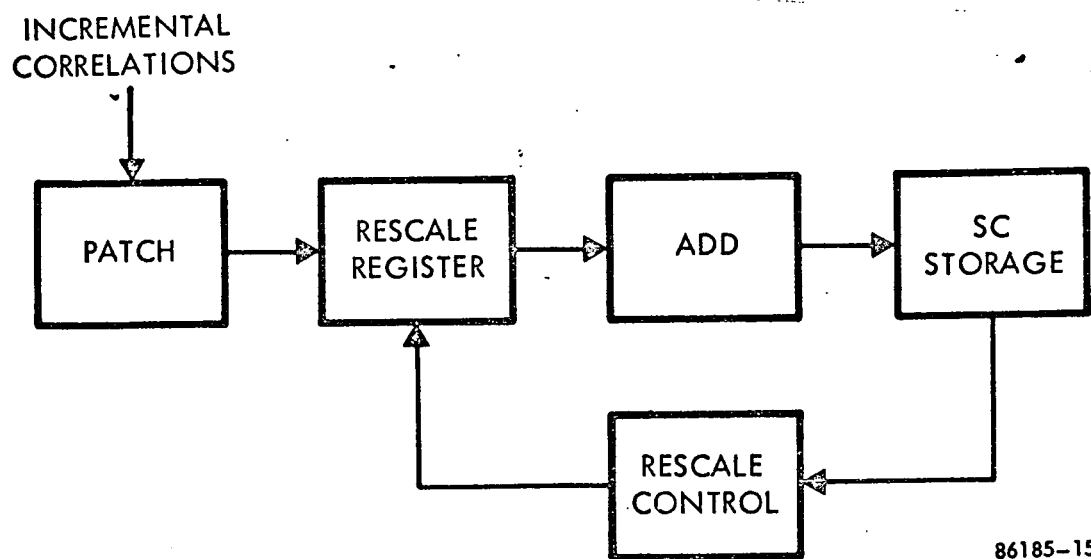
In the Block Diagram, the blocks labeled SC are 8-bit registers in which the largest survivor path correlations are stored. Sixty-four (64) of the largest survivor path correlations are increasing monotonic functions of time, thus, an overflow problem exists. When the largest survivor path has exceeded (127_{10}), the constant (128_{10}) is subtracted from the survivor correlation 6-bit times later. The survivor path correlations are all decreased by (128_{10}) and are prevented from overflowing the 8-bit SC registers. A 6-bit delay is necessary to ensure all SC registers have exceeded 127_{10} .

When the threshold (127_{10}) has been exceeded, the input to the rescale control is inhibited for 5-bit times so that the rescale control is not reactivated until the rescaling is completed. Reference may be made to the Rescale Loop Figure and Rescale timing chart (Figures A2-3.3.3.5-1 and A2-3.3.3.5-2).

A2-3.3.3.6 The Patch Function

The patch function shown in the Block Diagram allows any generator polynomials to be used for generation of the code. The 128 adders shown by the 64 SC registers may be grouped by pairs that receive the same incremental correlation independent of the code. This means that each of these 64 pairs of adders must receive one of the eight incremental correlations. The generator polynomials for the code determines which of the incremental correlations is routed to a pair of adders.

The patching for a code is accomplished by the use of plug-ins on a master board. Each code has associated with it a set of 40 prewired plug-ins. A code is implemented by inserting the plug-ins into the master board located on the logic chassis. The code can be changed by changing the plug-ins. Electrically, the patch function is shown in Figure A2-3.3.3.6. The vertical lines are incremental correlations of 5 bits each. Each of these correlations can go to any combination of the 64 pairs of adders. These lines can be connected to only one vertical line, i.e., they can be connected to only one incremental correlation.



86185-15

Figure A2-3.3.3.5-1. Rescale Loop

BIT TIME	SAMPLE IN RESCALE REGISTER	SAMPLE IN SC STORAGE
1	#1	
2	#2	#1 $128 \leq \text{MAX} \leq 148$
3	#3	#2 $128 \leq \text{MAX} \leq 169$
4	#4	#3 $128 \leq \text{MAX} \leq 190$
5	#5	#4 $128 \leq \text{MAX} \leq 211$
6	#6	#5 $128 \leq \text{MAX} \leq 232$
7	#7	#6 $128 \leq \text{MAX} \leq 253$
8	#8	#7 $0 \leq \text{MAX} \leq 146$

86185-16

Figure A2-3.3.3.5-2. Rescale Timing Chart

INCREMENTAL CORRELATIONS

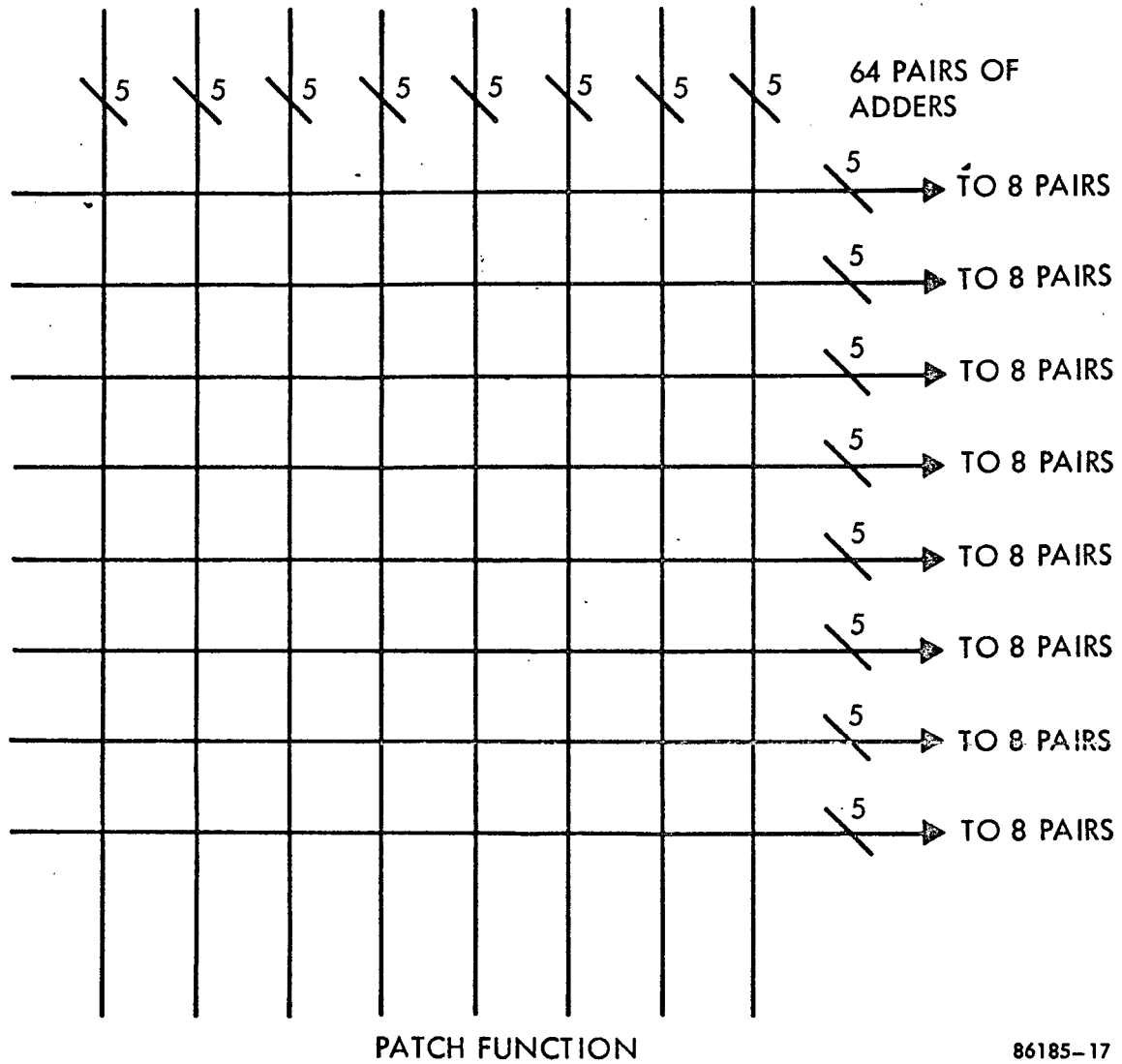


Figure A2-3.3.3.6. Patch Function

The above applies for rate $1/2$ codes except that there are only four incremental correlations. Each of the correlations will be present in two adders, thus distributing the loading over two adders rather than one.

The actual patching is accomplished by patching the MSB's of each incremental correlation on common plug-ins. The remaining bits are patched identical to the MSB's.

The operations to be performed following the patch function have an effect on the limiting rate of the entire decoder. Rather than allow the patch function delay to enter these sensitive operations and impose a limitation on rate, a storage register is placed after the patch function.

Patching is the same for shorter constraint length codes.

The following are the codes for which patch plugs are being provided with the system:

$K = 6, R = 1/3$	111001, 111011, 101101
$K = 5, R = 1/3$	10101, 11111, 11011
$K = 4, R = 1/3$	1111, 1101, 1011
$K = 6, R = 1/2$	110101, 100011
$K = 5, R = 1/2$	11001, 10111
$K = 4, R = 1/2$	1101, 1111
$K = 3, R = 1/2$	101, 111

Refer to Appendix A1 (Operation) for programming information.

It is necessary to store the running correlations for 2^{K-1} survivor sequences. This requires 2^{K-1} 8-bit registers. It is then necessary to add the incremental correlations to the 2^{K-1} survivor path correlations in such a way as to form 2^K sums. For each state in the state diagram (there are 2^{K-1} of these) a survivor path correlation is stored. A new correlation is computed for each state given a one is received, and also for each state given a zero is received. These 2^K sums represent 2^K sequences, hence it must be decided which of these 2^K sequences must be stored. This requires comparing two 8-bit numbers, determining the largest of the two numbers, and restoring the largest survivor path correlations in the SC register. Additionally, the results of the comparisons must be routed to register exchange gating. The maximum branch rate at which the decoder can operate is limited by the time required for the addition, comparison, and storage.

In order to operate at higher branch rates, the configuration shown in the Block Diagram is used. The survivor correlations corresponding to each of the 2^{K-1} states given a one was received and the survivor correlations corresponding to each of the 2^{K-1} states given a zero was received are stored in the SC register. The SC'_X registers store the correlation for the survivor path in the X state given a one was received and the SC_X registers store the correlation for the survivor path in the X state given a zero was received.

During each bit time, multiple operations are performed in parallel. First, out of the 2^K paths whose correlations are stored in the SC registers, the 2^{K-1} survivor paths are determined. In parallel, the 2^K paths each are added to their respective incremental correlations. This yields 2^{K+1} sums. These sums are the correlations for the 2^{K-1} survivor paths given any of the four possible two bit sequences are received. SC_0 contains the correlation for state 0 given a zero was received. The sum formed by the adder above SC_0 is the correlation corresponding to being in state zero, receiving a zero, and receiving another zero. The sum formed by the adder below SC_0 is the correlation corresponding to being in state zero, receiving a zero, and then receiving a one. SC'_0 contains the correlation for state 0 given a one was received. The sum formed by the adder above SC'_0 is the correlation corresponding to being in state zero, receiving a one, and then receiving a zero. The sum formed by the adder below SC'_0 is the correlation corresponding to being in state zero, receiving a one, and then receiving another one. In summary, we have for the zero state, the correlation with the received sequence, of the survivor path terminating in the zero state when the survivor path is followed by a 00, 01, 10, 11 sequence of information bits.

A similar calculation is made for each of the other 2^{K-1} states when the survivor path is followed by 00, 01, 10, or 11 sequence of information bits. These 2^{K+1} sums are presented to the comparison logic which eliminates paths.

The virtue of this configuration is that rather than performing the addition and comparison sequentially, we are performing the operations simultaneously. This allows a much greater information rate through the decoder.

When a shorter constraint length code is used, the SC registers will contain more than one survivor correlation path for each state given a received bit. Thus for $K = 5$ two SC will contain survivor correlation paths for each state.

One will be the maximum correlated path. The number of paths terminating in a state will double for every decrease in the constraint length.

A2-3.3.3.8 Comparisons of Correlations and Elimination of Paths

There are 32 comparators which receive the contents of the SC registers. Recall that the 64 SC registers contain 2 numbers of each of the 32 survivor paths. The number contained in the SC_0 register is the correlation for the survivor path terminating in state 0 given a 0 bit was received. Since there are only 32 states, certain pairs of SC registers represent correlations for paths terminating at the same state. The path with the largest correlation is decided by comparing the two SC registers and the result is stored. The output of each comparator is a signal which indicates whether the upper or lower input to the comparator was largest.

Meanwhile, the adders have formed 128 sums, 64 of these sums however, are correlations for extensions of nonsurvivor paths which are discarded. The remaining 64 sums are correlations for the 32 new survivor paths given a one was received and correlations for the 32 new survivor paths given a zero was received. These quantities are stored in the correct SC registers. The routing of these sums is by the selection logic, which is located with the SC registers.

Suppose that a comparator has as its inputs the SC_y and SC'_y , and both of these paths terminate in state X. If SC_y is greater than SC'_y , the sum out of the adder above SC_y is stored in SC_x . The sum out of the adder below SC_y is the correlation for the path terminating in state X with a one as the next bit and is stored in SC_{x+1} . Conversely, if SC'_y is greater than SC_y , then the sum out of the adder above SC'_y is the correlation for the path terminating in state X with a zero as the next bit and is stored in SC_x . The sum out of the adder below SC'_y is the correlation for the path terminating in state X with a one as the next bit and is stored in SC'_{x+1} .

In summary, during each bit time the 64 paths whose correlations are stored in the SC registers are compared in pairs (pairs which terminate in the same state) to determine the survivor paths. Simultaneously, for each of the 64 paths whose correlations are stored in the SC registers, the correlations for 128 one bit extensions of these paths are calculated. When the 32 surviving paths are determined, this information is used to gate the correlations for the 64 one bit extensions of these paths into the SC registers.

The important feature of this implementation is that an 8-bit addition and a comparison of two 8-bit numbers are performed simultaneously rather than sequentially.

The SC registers are 8-bit registers whose contents are bounded by zero and 255_{10} . The inputs from the patch function are numbers from 0 to 21_{10} depending on the value of the incremental correlation. The sums produced by adders will lie between zero and 255_{10} .

Since the SC register are always positive numbers, the comparison can be made by adding the upper number and the one's complement of the lower number. The carry output, from the adder determines whether the upper or lower SC register (of the pair feeding the comparator) is the largest.

The comparator output, in addition to comparing the correlations and eliminating nonsurvivor paths, also indicates what the bit is in the survivor path. If the upper SC register is the largest, then the survivor path contains a zero in the branch most recently decided. If the lower SC register is the largest, then the survivor path contains a one in the branch most recently decided.

A2-3.3.3.9 Register Exchange Logic

The register exchange logic consists of thirty two (32) survivor sequence registers each 32 bits long. The loading of the survivor sequence registers are controlled by the comparing of the survivor correlation registers. If $SC_0 \geq SC'_0$ then a zero is loaded into the first bit of SS_0 and the contents of SS_0 are loaded into SS_0 , but shifted by one bit. If $SC_0 < SC'_0$ then a one is loaded into the first bit of SS_0 and the contents of SS_{16} are loaded into SS_0 , but shifted by one bit. The remaining SS registers work similarly to SS_0 . The inputs to the registers are represented by the trellis diagram (Figure A2-3.3.3.9). The upper branch into a state occurs when a zero is the result of the comparator. The lower branch into a state occurs when a one is the result of the comparison.

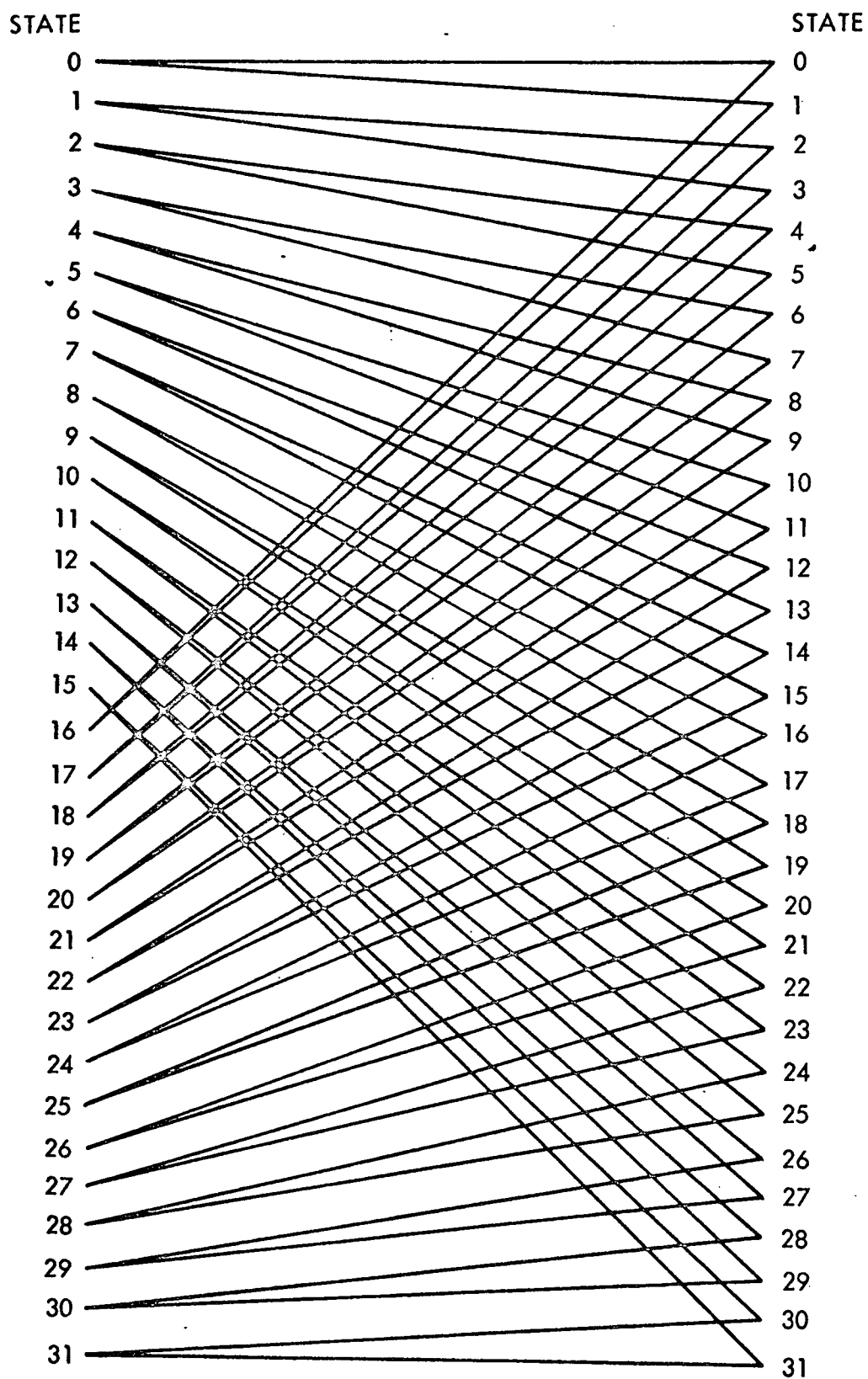
There is a two bit delay between the comparison and the loading of the SS registers. This delay is required because the comparators and SS registers are located in separate logic drawers. One bit of delay is on each drawer to prevent the delay of the cabling from limiting the speed of operation.

A2-3.3.3.10 Majority Bit Decision Logic

The majority bit decision logic receives bits from each SS register. These bits correspond to various depths of the sequence. Eight different depths can be selected. They are 4, 8, 12, 16, 20, 24, 28, or 32. These are programmed by using switches located in the logic.

The majority bit decision logic sums the number of ones at the selected depth. This summing process requires four branch times. The decoder output is a logical one if the sum is 16 or greater.

The decoder output is available to the group sync and front panel.



86185-18

Figure A2-3.3.3.9. Trellis Diagram

The delay through the Viterbi Decoder varies depending on decoding depth. For rate $1/3$ there is a fixed delay of 16 bits plus a variable delay of 0 to 28 bits in increments of 4 bits. For rate $1/2$ there is a fixed delay of 17 bits plus the variable delay.

Three binary weighted lines are sent to the Bit Comparator A for determining the variable delay required in the comparator.

A2-3.3.3.11 Variable Parameter Implementation

The Viterbi Algorithm Decoder has the feature that code parameters can very simply be changed. These variable parameters allow testing of a wide variety of codes. This paragraph discusses the implementation of these parameters.

The decoder is capable of handling codes with constraint length up to 6. Functions are required to change for the various constraint lengths. These are the patching, bandspread ratio, quantization, constraint length, and decoder output depth.

The generator polynomials for the codes are changed by replacing the patch plugs. Any generator polynomial, maximum constraint length 6, can be implemented using the patch plugs.

The threshold and check depth for the branch synchronizer must be changed for the constraint lengths. Two switches in the logic will control the selection. The thresholds are 60 for $K = 5$ or 6 and 30 for $K = 3$ or 4. The check depths are 32 for $K = 6$, 27 for $K = 5$, 22 for $K = 4$ and 16 for $K = 3$.

An additional patch allows the constraint span to be varied. A digital network is provided which will allow the inputs to the majority bit decision logic to come from any fourth staged of the SS registers. The enabling is controlled by three switches located in the logic. Decoder performance is a function of the decoder constraint span. The test set has the capability of quantitatively determining the effect.

The decoder is capable of handling one, two, or three bit quantization. The only logic change required is to force the unused bits of the input register to add equally to all incremental correlations. This is controlled by two switches located in the logic.

A front panel switch controls the bandspread ratio. For rate $1/3$ codes, the decoder is used as in the Block Diagram Figure 3.3-2. For rate $1/2$ only two 3-bit words are received for each branch. The output lines from the third 3-bit portion of the input registers are arranged so that they add equally to all of the incremental correlations. This is performed logically, driven from the front panel switch. From this point on, no logical modifications are required.

A2-3.3.3.12 Interface

The Viterbi receives data and clocks from the bit sync and outputs the MSB of the data and the clocks to the Fano Decoder. The decoder outputs error corrected data to the group synchronizer. A clock in phase with the output data is provided to the group synchronizer.

The Viterbi receives enables for the bandspread ratio from the front panel. It outputs enables for variable decoder output depth.

A2-3.4 Fano Decoder - A8A1A9 (A13-A22)

The Fano Decoder performs the following functions:

- a. Accepts serial data and clock (to 18 MHz) from bit synchronizer for rate 1/2 systematic codes with constraint length and polynomial terms to 40.
- b. Establishes polarity and branch synchronization on the input serial data.
- c. Provides polynomial setup to match the encoder.
- d. Utilizing generated syndromes and the Fano Algorithm, provides correction of the data bit stream.
- e. Outputs corrected data and clock at rates up to 9 MHz.

A2-3.4.1 Theory of Operation

The paragraphs of Section A2-3.4.1 supply general information about codes as a basis to understanding the Fano Decoder. Details about implementation follow in Section A2-3.4.2.

A2-3.4.1.1 General Information

Consider the polynomial, $I(x)$, to be a sequence of information bits coded by some generator polynomial, $G(x)$, to form the transmitted sequence, $T(x)$. This is expressed by:

$$T(x) = G(x)I(x)$$

For example, let the binary number, 11010, represent the coefficients in the polynomial

$$I(x) = 1 \cdot x^0 + 1 \cdot x^1 + 0 \cdot x^2 + 1 \cdot x^3 + 0 \cdot x^4$$

and let binary number 11 represent the coefficients in:

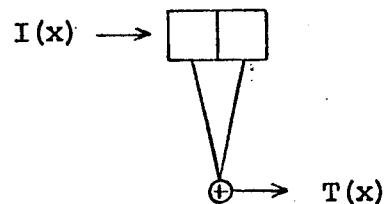
$G(x) = 1 + x$. Then $G(x)I(x)$ becomes:

$$\begin{aligned}(1 + x)(1 + x + x^3) &= 1 + x + x^3 + x + x^2 + x^4 \\ &= 1 + x^2 + x^3 + x^4 \quad (x + x = 0)\end{aligned}$$

Writing only the coefficients of $T(x)$ as its binary representation gives:

$$T(x) = 101110$$

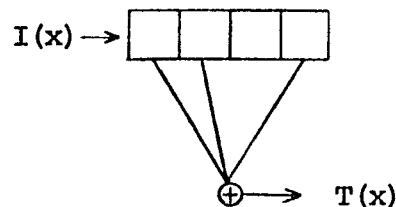
This product can be produced by feeding $I(x)$ into a shift register and generating the Mod 2 sum, $T(x)$, with inputs to the Mod 2 adder corresponding to the terms of $G(x)$.



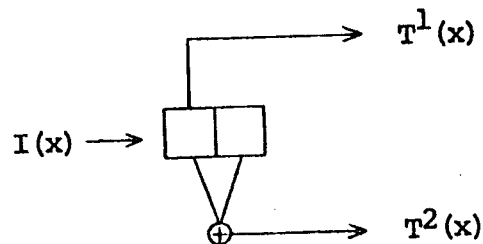
Assume the register to contain zeros to begin with and that the bits 11010 shift through the register MSB first. The output of the Mod 2 Adder will then be the sequence, $T(x) = 101110$, as can be demonstrated by a step-by-step calculation. Thus the configuration shown forms the desired multiplier.

To apply any generator polynomial to $I(x)$ one need only to shift $I(x)$ through a register and Mod 2 add the outputs of the appropriate stages. Stage zero (input stage) supplies to the Adder the coefficient of the zero order bit; stage 1 (second from the left) supplies to the Adder the coefficient of the 1st order bit; etc.

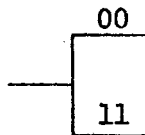
For example, if $G(x) = 1 + x + x^3$, connect the Mod 2 Adder as follows:



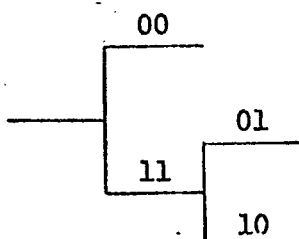
Now consider a configuration where $G^1(x) = 1$ and $G^2(x) = 1 + x$:



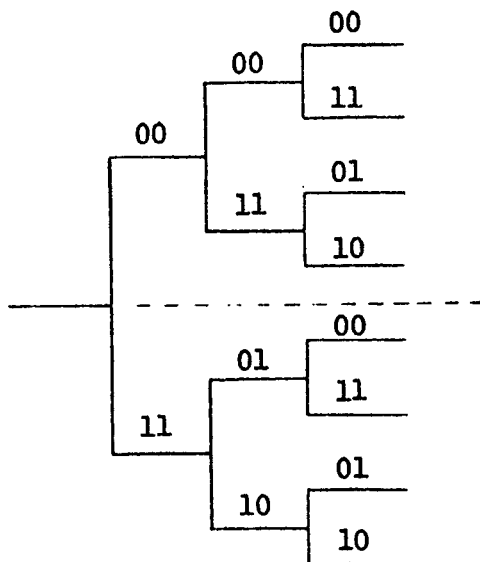
As information bits enter the register, pairs of transmit bits are generated. Assuming the register to be in its initial state (a, 0 in the first stage) one can construct a code tree of possible sequences as follows. When a "0" enters the register transmit bits, $T^1, T^2 = 0, 0$ are generated; had a "1" entered, one would have $T^1, T^2 = 1, 1$. These possible changes from the initial state form the first branches:



Using these as starting points, the next branches can be formed. When $T^1, T^2 = 1, 1$ the register contains bits 10; shifting in a "0" results in $T^1, T^2 = 0, 1$; had a "1" entered, one would have $T^1, T^2 = 1, 0$. The added branches extend the tree:



Continuing this process, making the upper branch of any pair of branches represent T^1, T^2 for a zero entering the register adds to the tree as below:



Note that for a 5-bit sequence sent through the register, one obtains a 10-bit output sequence. The 5-bit input sequence can have 32 distinct values; the output 10 bits can have 1024 distinct values. Now only 32 of the 1024 output code words correspond to the 32 possible input sequences; the others being "unallowed." If the 32 possible output code words are plotted, each is different from its neighbors in some number of bit positions. The important thing to note is the least separation between allowed code words and in particular the least separation of allowed code words located in the top half of the tree from those in the bottom half of the tree.

* For the tree above, the least separation is 3. As an illustration, select any code word from the top half, e.g., 000000, and any code word from bottom half, e.g., 110100, and Mod 2 add them:

$$\begin{array}{r} 00\ 00\ 00 \\ \underline{11\ 01\ 00} \\ 11\ 01\ 00 \end{array}$$

The two code words differ by three 1's or 3 digits.

Any other pair of code words from opposite halves of the tree will differ at least as much. This difference is referred to as the hamming distance. The minimum hamming distance, d , is related to the number of errors (erroneous digits) which can be corrected, t , by

$$d = 2t + 1.$$

That is, one could introduce a single error in any of the 6-bit allowed code words above and still find the new unallowed sequence closer to the original code word than all other allowed code words.

A decoder can now be discussed which essentially uses the code tree to correct errors found in the transmitted sequence.

The decoder introduced here differs from the one implemented in the Test System but will serve as a good basis for understanding it.

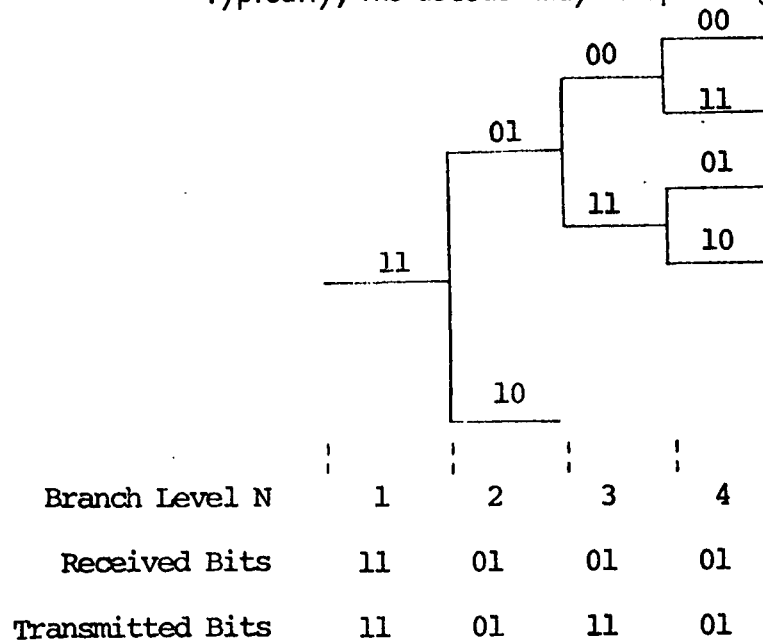
The description of the decoder's operation is mainly imparted by the use of three examples. The first example supplies the most general information and serves to introduce a few terms needed. The second example dwells on modes mentioned by the first. A third example is given to illustrate more comprehensive operation. This example is an extension of the first one. Finally, a flow chart is given which later becomes the basis of the decoder implemented in the Test System.

Consider now the first example. Suppose a pair of transmitted bits, T^1 , T^2 arrive at the decoder. From the previously received pair, the decoder decides whether the new bits extend

the tree. If the bits were expected, they will be used to determine what future pairs are allowable, and in this way, the decoder moves one branch further into the tree. If the transmitted bits do not form an extension to the code tree, the decoder may or may not move forward. When such a situation occurs, two avenues are open. The decoder may decide to go forward by hypothesizing a correction for the unexpected bits. This will allow the decoder to proceed one branch further into the tree and allow it to examine another new pair of bits. The correction introduced may have indeed been the right one, properly predicting the pair following the one containing errors, and allowing the decoder to move on, processing new bits and continuing through the tree.

A second avenue open to the decoder when an unexpected pair arrives is to recall a previously processed pair, assume it to be incorrect and hypothesize a correction. The correction here may be one which moves the decoder from an upper branch to a lower branch. In this case, new allowable pairs are predicted, perhaps including the pair which originally appeared in error.

Typically, the decoder may be operating at branch level 1. From the



illustration, note the pair 11 is received. This was the pair transmitted so the decoder is set to move to branch level 2. At this point, bits 01 or 10 are expected; pair 01 is received. So, again the decoder is set to move forward. At level 3, bits 00 or 11 are expected. The illustration shows pair 01 is received. At this time, the decoder must decide on one of the two possible actions.

A correction of the error may be attempted. Suppose that when this is done, the decoder always hypothesizes trying the upper branch first. Here then, the decoder would change the 01 pair received at level 3 to 00. With this hypothesis, the decoder would expect pair 00 or 11 at level 4. From the illustration, the pair actually received is 01. Thus a second

error would have to be assumed to continue on this path. But let the decoder take the second avenue of approach, and back up one branch level to 3. The pair 01 received at that level was originally presumed to be 00. This time the decoder will try 11 as a correction. Now pair 01 or 10 is acceptable for level 4. Pair 01 is received. So, the decoder is on its way further into the tree.

Some mechanism must exist in the decoder which tells it to choose to move through an error or to back up. In the Test System Decoder, a number called the Metric Value controls all movement. When the pair of bits being examined would extend the tree, the number will be incremented by one, and the decoder will always move forward. When the newly arrived pair disagrees with the expected allowable pairs, the number may be decreased and the decoder moved forward. This is the case where the decoder hypothesizes a correction. Whenever the Metric becomes negative, however, a backward move is required.

In the illustration, suppose the Metric Value is $\Delta - 2$ at $N = 1$, and suppose the single errors have a weight of $-\Delta + 1$, and double errors a weight of $-2\Delta + 1$. The following table shows the behavior of the Metric Value, both for a generalized case, and for the specific case of $\Delta = 5$. When the decoder moves from level 1 to level 2, pair 01 is expected and the Metric Value changes to $L_2 = \Delta - 1$. Moving to level 3, the error is absorbed. A correction of 01 to 00 is assumed, and $L_3 = 0$. A move is made to level 4. Since bits 00 or 11 are expected and bits 01 are received, $L_4 = -\Delta + 1$. For the $\Delta = 5$ case, L_4 is negative. A required backward move now resets the Metric to $L_3 = 0$; a single error is absorbed when a new correction hypothesizes 01 at 3 should be 11. This correction leads the decoder to expect the pair 01 it receives at level 4 and L_4 increases to +1.

Tree Level	Error Assumption	Next Move Direction	Add to Metric	General Metric	Actual Metric For $\Delta = 5$
1	None	Forward	1	$\Delta - 2$	3
2	None	Forward	1	$\Delta - 1$	4
3	One	Forward	$-\Delta + 1$	0	0
4	One	Backward	$-\Delta + 1$	$-\Delta + 1$	-4
3	One	Forward	$-(\Delta + 1)$	0	0
4	None	Forward	1	1	1

Note how the Metric increases only slowly as the decoder processes pairs having no errors and decreases sharply with errors. For $\Delta = 5$, the Metric quickly decreases by -4 for single errors and $-2\Delta + 1$ or -9 for double errors.

In the bit checking operation, it is important to keep a positive Metric value relatively close to some threshold number, i.e., as progress is made through the tree and the Metric value grows larger, it is important that a threshold value be updated along with the Metric value so that discovery of certain error conditions requires the Metric drop below this threshold and thus alert the decoder.

The Metric to threshold relationship can be satisfied by defining the threshold to be zero and representing the Metric by using a counter which advances to some acceptable value and then rolls over to start again. By choosing the rollover point to be small, almost any error condition will require violation of the threshold if the effect of that condition were actually added to the Metric.

Suppose the Metric counter contained L_{N-1} rather than L_N . Then, for this example, the Metric could be a simple Mod-5 up-down counter. By allowing the counter to store L_{N-1} , it never needs to assume negative values. When an error is found, the decision to move forward just requires L_{N-1} to be greater than or equal to the error's weight.

Notice how this example was idealized to allow an easy correction of the error at branch 3. L_2 was large enough to absorb the error. A single backward move from level 4 and switching from the upper to the lower path at level 3 was all that was required to make the correction. Suppose L_2 had been some value less than $\Delta - 1$. With the Mod-5 counter suggested as the Metric accumulator and with the criteria that L_{N-1} be positive as the decoder proceeds into the tree, progress would stop at level 3. This problem is discussed below.

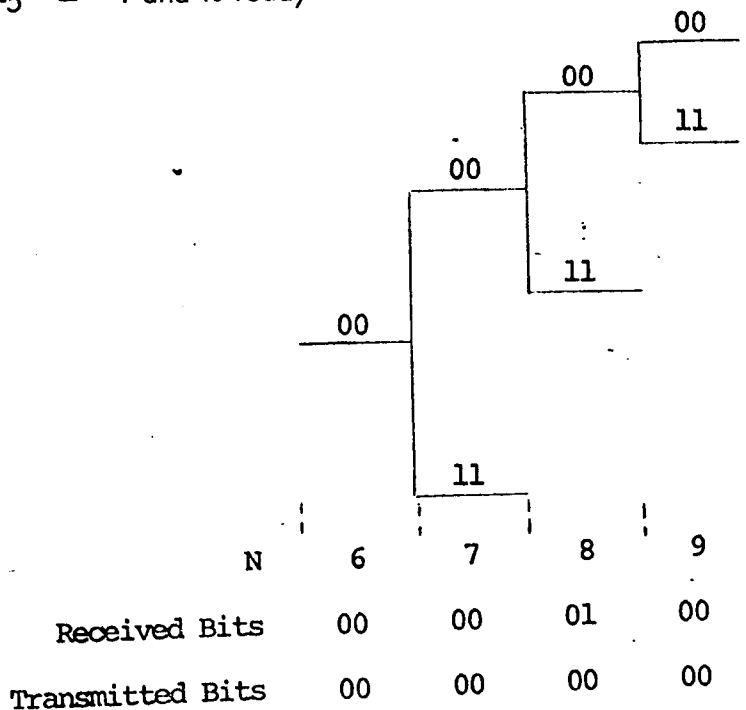
At this point, consider a second example. A procedure called lowering the threshold will be introduced. With this technique, the decoder, faced with an error situation impeding all forward progress, adds an amount, Δ , to its Metric and reinitiates its search.

The term lower-the-threshold may sound misleading in the sense that the threshold was earlier defined to be zero and this procedure suggests changing the definition. Actually, threshold, T , remains zero with respect to the value in the Metric counter. When Δ is added to the Metric Value, T , still zero, is then Δ further below the new Metric and in this sense lowered.

Threshold lowering here will be accomplished by changing the Mod-5 counter used earlier. Assuming $\Delta = 5$, the Mod-5 counter is a Mod- Δ counter; extending its length with additional stages gives a Mod- Δ counter in series with a Δ counter. Lowering the threshold by adding Δ then can be accomplished by incrementing the added counter by one count.

With the modification suggested for the Metric, the decoder will try to move forward using only the Mod- Δ counter. As the decoder moves through error free branches the Mod- Δ counter adjust increments to $\Delta - 1$ and rolls over, updating the threshold as before and effectively raising it. Only when hope for further forward progress is impaired does the decoder use the new Δ counter.

Typically, the decoder may be operating at branch level 6. From the illustration below, note pair 00 is received. Let $L_6 = 0$, i.e., at level 6, the Mod- Δ counter contains $L_5 = \Delta - 1$ and is ready to roll over.



Bits 00 are received at level 7. So, $L_7 = 1$. Bits 01 at level 8 contain a single error. Since the Metric counter at level 8 contains $L_7 = 1$, the decoder is directed to back up. A shift to level 7 follows. There the decoder assumes bits 00 received at this level are in error. Changing this assumed error of 00 to 11 and moving forward along the lower path of 7 would require the Metric to absorb $-2 \Delta + 1$. Since the Metric contains $L_6 = 0$, a second reverse shift would be ordered.

If no new mechanism was introduced here, the decoder would continue backing and searching lower paths - a fruitless task. The Fano algorithm calls for a threshold lowering when $L_N < T$ and when $L_{N-2} < T$. In this example $L_8 < T$ but $L_6 = T$; so, the reverse move to level 7 should occur. This move requires searching the lower path and results in L_7 being negative. Now, since the threshold was raised at level 6, L_5 is negative and a lowering should occur. This means L_7 should be changed from its old value of 1 to a new value of $1 + \Delta$. Further, the decoder must move forward along the upper path first used at level 7. Now the decoder is able to move through the error at level 8 where $L_8 = 2$, and then to level 9, etc.

In summary, note that the decoder keeps the threshold low enough to do a complete search of paths. Almost any error condition will cause the decoder to back up and test a different branch. When possibilities are exhausted, the threshold is lowered by Δ . This allows the decoder to again initiate its search but, with the less sensitive threshold, the decoder will be allowed to move a little further into the tree before discarding a selected path. The less sensitive threshold thus allows the decoder to ignore the first error as a given path is checked.

The symbol I_N is introduced here to distinguish between upper and lower paths at level N . Since the upper path was arbitrarily selected for a first attempt through a branch in the examples $I_N = 0$ will mean use-upper-path. This concept will be modified later.

The examples used have a few limitations. For one, they ignore the case where $N = 1$ and where a reverse shift is not allowed. $N = 1$ signifies no older branches are available.

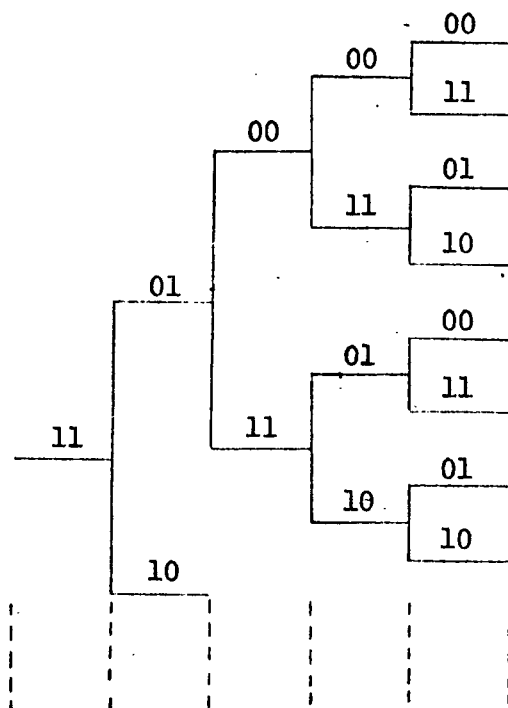
As a second limitation, consider the possibility of the decoder moving forward on a lower path after discarding the upper one. The decoder may be required to back up into such a branch level and must not repeat searching the upper path. A new symbol, H , is introduced here to alert the decoder to avoid the unnecessary step.

Consider now the third example. Referring to Table A2-3.4.1.1, the decoder tries the upper path at branch 3, fails, tries the corresponding lower path, succeeds and proceeds to branch 4. At branch 5 a disagreement is found which would cause the Metric to be negative; the decoder backs up. At level 4 the lower path is tested. This test requires the Metric to be negative so the decoder again backs up. At level 3, both paths have been checked; $H = 1$. So, again the decoder backs up. At level 2, the lower path is tested, etc.

Finally, in this case, the decoder reaches level 1 and, because this is the end of the bit sequence available, the threshold is lowered and forward moves are made. Had the limitation of $N = 1$ not caused a reversal, the condition, $L_N, L_{N-2} < T$, eventually would have and, as in this example, forward moves would have followed.

By step 17, the decoder has passed through two errors and is headed forward. If no further errors are discovered, the Metric will build up to a value of 4 by step 19. At this point, the threshold will be raised by just letting the Metric roll over to zero on step 20.

A flow chart of the Fano algorithm is given in Figure A2-3.4.1.1.



N: 1 2 3 4 5
 Received: 11 01 01 01 01
 Transmitted: 11 01 00 11 01

Table A2-3.4.1.1. Example of Fano Decoder Operation

Step	N	L_N	Received Bits	Bits Assumed	H	I_N	Comment
1	2	4	01	01	0	0	
2	3	0	01	00	0	0	
3	4	-4	01	00	0	0	back up
4	3	0	01	11	0	1	
5	4	1	01	01	0	0	
6	5	-3	01	00	0	0	back up
7	4	-9	01	10	0	1	back up
8	3			-	1	1	back up
9	2	-6	01	10	0	1	back up
10	1	-7	11	00	0	1	$T \rightarrow T - \Delta$
11	1	8	11	11	0	0	
12	2	9	01	01	0	0	
13	3	5	01	00	0	0	
14	4	1	01	00	0	0	
15	5	-3	01	00	0	0	back up
16	4	1	01	11	0	1	
17	5	2	01	01	0	0	

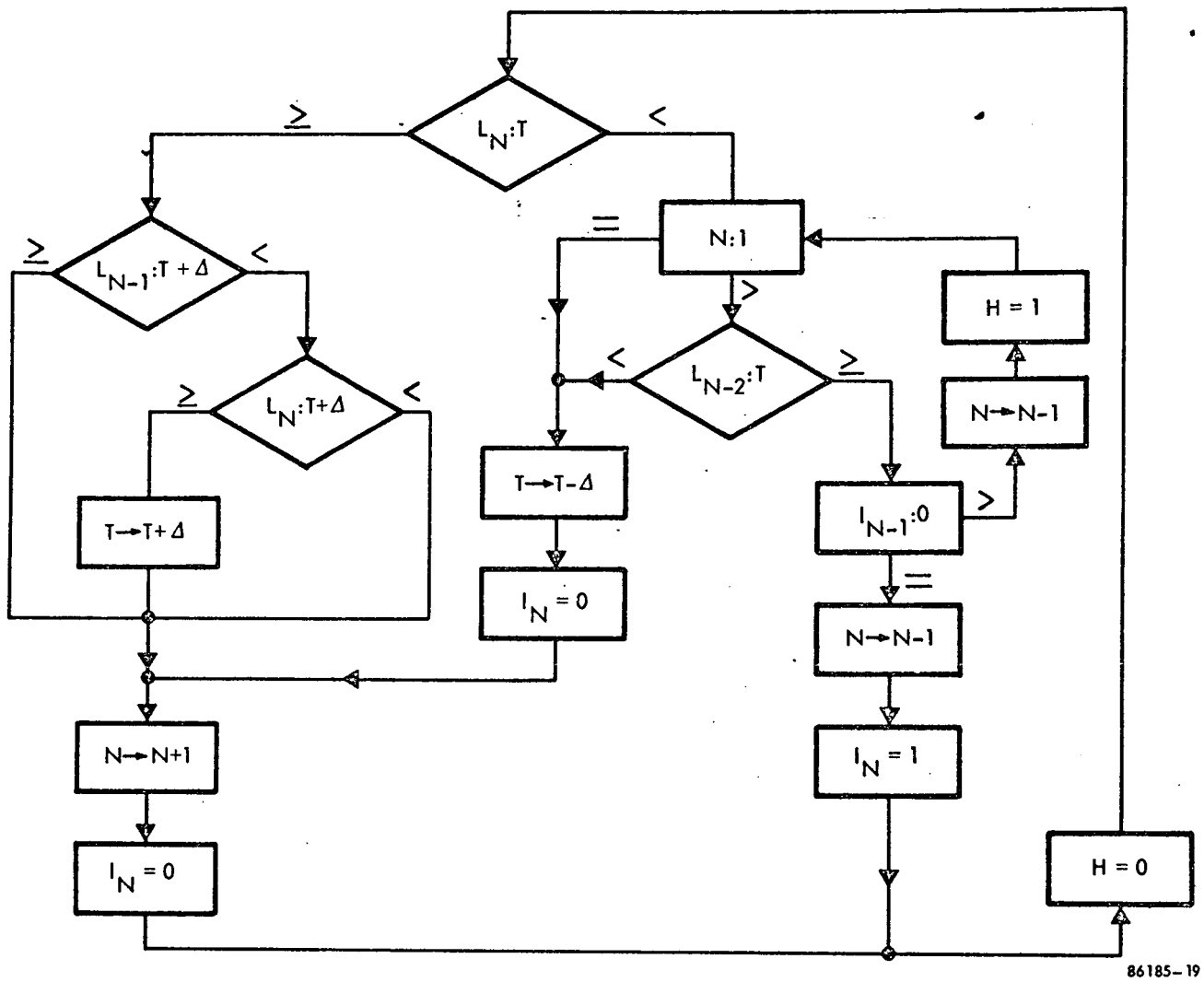


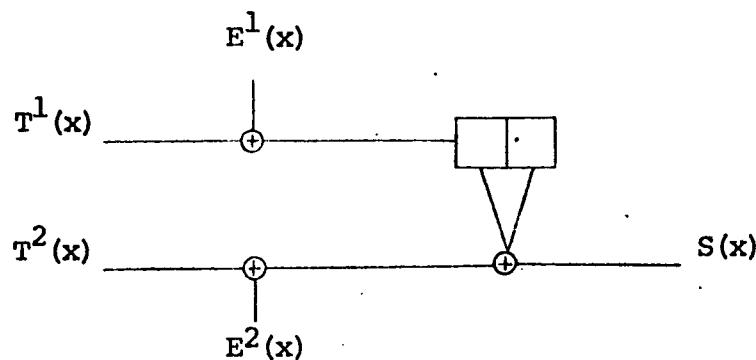
Figure A2-3.4.11. Fano Decoder Flow Chart

A2-3.4.1.2 Detail Description

A variation of the above described decoding technique can now be discussed. This technique differs from the above in that bit pairs T^1 , T^2 are processed to form syndrome bits and then the syndrome bits are used to make decisions about movement through the tree.

As a first step, consider the relationship of errors to the data received. The bits $T^1(x)$ are coefficients of a polynomial; corresponding to each of these coefficients is a coefficient for an error polynomial, $E^1(x)$. Where a transmitted bit is changed, its corresponding coefficient in the error polynomial is a one. Where a transmitted bit remains unchanged, its corresponding coefficient bit in the error polynomial is a zero. The received bits then are the Mod-2 sum of the $E(x)$ and $T(x)$ coefficients.

The syndrome bits are polynomial coefficients generated when the Mod-2 sum, $T^1(x) + E^1(x)$, is multiplied by the generator polynomial used by the transmitter and the resulting bits Mod-2 added to the polynomial $T^2(x) + E^2(x)$ as shown below.



Note that since

$$S(x) = [T^1(x) + E^1(x)] g^2(x) + [T^2(x) + E^2(x)]$$

$$T^1(x) = I(x),$$

and $g^2(x) = (1 + x),$

$$T^2(x) = I(x) g^2(x) = I(x) (1 + x)$$

Then

$$\begin{aligned} S(x) &= \left[I(x) + E^1(x) \right] (1+x) + \left[I(x) (1+x) + E^2(x) \right] \\ &= I(x) (1+x) + I(x) (1+x) + E^1(x) (1+x) + E^2(x) \end{aligned}$$

In the Mod 2 arithmetic used here,

$$I(x) (1+x) + I(x) (1+x) = 0,$$

$$S(x) = E^1(x) (1+x) + E^2(x),$$

showing that the syndrome bits are independent of the data.

$$\text{Writing } E^1(x) = e_0^1 + e_1^1 x + e_2^1 x^2 \dots \text{ and}$$

$$E^2(x) = e_0^2 + e_1^2 x + e_2^2 x^2 \dots,$$

$$\begin{aligned} S(x) &= (e_0^1 + e_0^2) + (e_0^1 + e_1^1 + e_1^2) x \\ &\quad + (e_1^1 + e_2^1 + e_2^2) x^2 + \dots \\ &= S_0 + S_1 x + S_2 x^2 + \dots \end{aligned}$$

The coefficient of x^0 here is the Mod-2 sum $(e_0^1 + e_0^2)$; it corresponds to the x^0 coefficients of $T^1(x)$ and $T^2(x)$. If $e_0^1 = e_0^2 = 1$, then their Mod-2 sum of zero indicates an error in both bits T_0^1, T_0^2 . If $e_0^1 = e_0^2 = 0$, their Mod-2 sum of zero indicates no errors. If the Mod-2 sum of e_0^1 and e_0^2 is 1, a single error exists in bits T_0^1, T_0^2 .

Note at this time that an error in bit T_0^1 expressed by $e_0^1 = 1$ is carried over into S_1 . Similarly, an error, $e_1^1 = 1$, in bit T_1^1 carries over into S_2 , etc. If the generator polynomial has been $1 + x + x^3$, e_0^1 would appear in S_0, S_1 and S_3 ; if the generator polynomial had N terms, e_0^1 would appear in each of the younger syndrome bits which corresponded to a nonzero coefficient in the N term polynomial. This carryover of error coefficients into neighboring sums has the effect of inverting them; $e_0^1 = 1$ enters the Mod-2 sum

$$S_1 = e_0^1 + e_1^1 + e_1^2 x$$

and inverts S_1 .

Using the properties mentioned thus far, the decoder examines the individual syndrome bits and generates the coefficients of the polynomials, E^1 and E^2 . The coefficients of E^1 are Mod-2 added to corresponding coefficients of the received polynomial ($E^1 + T^1$). This then yields the transmitted polynomial T^1 :

$$E^1 + (E^1 + T^1) = T^1.$$

As a first step in the process, syndrom bit, S_N , is examined. Coefficients E_N and P_N of polynomials E^1 and E^2 are then hypothesized and shifted into holding registers. As this shift occurs, the next syndrome bit, S_{N+1} , is moved forward.

Just what is hypothesized depends on the state of S_N and whether the given syndrome bit had been previously examined. Quite often the E_N, P_N pair hypothesized is incorrect; a reverse shift follows, and a second pair is hypothesized. This behavior parallels the branch switching of Paragraph A2-3.4.1.1. There an attempt was made to correct an error by first trying the upper branch; when this failed, movement along the lower branch was attempted if the Metric permitted.

The symbol, I_N , used in Paragraph A2-3.4.1.1 is used here. I_N will be set to a one whenever a reverse move occurs. If $I_N = 0$ when the decoder examines S_N , the decoder predicts the most probable E_N, P_N pair. That is, for $S_N, I_N = 0, 0$, the pair $E_N, P_N = 0, 0$ is hypothesized rather than the less probable pair, $E_N, P_N = 1, 1$. For the case $S_N = 1, E_N, P_N = 1, 0$ or $0, 1$ are equally probable and $1, 0$ is arbitrarily assigned the "more probable" status. So, for $S_N, I_N = 1, 0$, pair $E_N, P_N = 1, 0$ is hypothesized.

The Metric here is the same as used earlier. When the decoder has $S_N, I_N = 0, 0$, it hypothesizes no errors and increments the Metric by one. A forward move with $S_N, I_N = 0, 1$ requires hypothesizing two errors and debiting the Metric by $-2\Delta + 1$. A hypothesis of $E_N, P_N = 1, 0$ or $0, 1$ requires debiting the Metric by $-\Delta + 1$. Analogous to its behavior in Paragraph A2-3.4.1.4, the decoder can move forward as long as L_{N-1} contained in the Metric is equal to or greater than the error condition represented by S_N, I_N .

The same threshold lowering device is needed here that was used earlier. In Paragraph A2-3.4.1.1, threshold lowering required the decoder to reinitiate its search with the upper branch - disregarding I_N ; similarly, here the decoder must reinitiate its search with the most probable pair, $E_N, P_N = 0, 0$ or $1, 0$ depending on S_N .

Whenever the decoder hypothesizes $E_N = 1$, it is implying the younger syndromes are affected. In particular, the hypothesis implies those syndromes corresponding to the nonzero terms of the generator polynomial have been inverted. Mod-2 adding the polynomial to the younger syndromes will invert them a second time and hence, restore them to their proper state. Note, however, that a mistaken hypothesis and polynomial addition will wrongly invert the younger syndromes and introduce more errors (syndrome "1"s). Discovery of this mistake will require a reverse shift, a new hypothesis, $E_N = 0$, and a second polynomial addition reversing the effect of the first.

In the examples above, the generator function most often just contained two terms or had a constraint length of 2. In the test set, the constraint length is 40. To implement the Mod-2 addition required, a special 40 stage register is used. The register is constructed so that the polynomial may be added in between forward or reverse shifts. When no corrections are required, this unit (called the computation register) performs like an ordinary shift register.

The basic configuration relating the Computation Register (CR), Error Register (ER) and Parity Register (PR) is shown in Figure A2-3.4.1.2-1.

In the Forward Mode, Error bits, E_N , and Parity bits, P_N , are generated by the control logic from the Syndrome bit, S_N , and state flip-flops. Once the forward shift occurs, E_N and P_N are loaded into the first stages of the ER and PR and become bits E_{N-1} and P_{N-1} . Coincidentally, S_N is replaced by S_{N+1} .

In the Reverse Mode, S_N is regenerated by the control logic from bits E_{N-1} and P_{N-1} . Once the reverse shift occurs, S_N generated from E_{N-1} and P_{N-1} replaces the S_N present at the output of the CR. Coincidentally, E_{N-1} and P_{N-1} are replaced by E_{N-2} and P_{N-2} , respectively.

From the discussion given thus far, E_{N-1} will be set to a one when

$$\left[(S_N \oplus I_N) \cdot (\overline{T \rightarrow T - \Delta}) + S_N \cdot (T \rightarrow T - \Delta) \right] \cdot \text{Fwd} + E_{N-2} \cdot \text{Rev} = 1;$$

(where " \cdot " is the logical AND and "+" is the logical OR)

P_{N-1} will be set to a one when

$$I_N \cdot (\overline{T \rightarrow T - \Delta}) \cdot \text{Fwd} + P_{N-2} \cdot \text{Rev} = 1.$$

These two equations will now be used as a starting point in defining the decoder control logic. The discussion below will cover the generation of $(T \rightarrow T - \Delta)$, Fwd Poly EN, Fwd and Rev.

The discussion of threshold lowering is covered in two parts; $T \rightarrow T - \Delta$ is found for clock period N during which it is used. Attention is then given to predicting $T \rightarrow T - \Delta$ a clock period early. This is done in steps. Each of the terms of the equation for $T \rightarrow T - \Delta$ is considered separately. Then, the predictions of the individual terms are combined to give the expression predicting $T \rightarrow T - \Delta$.

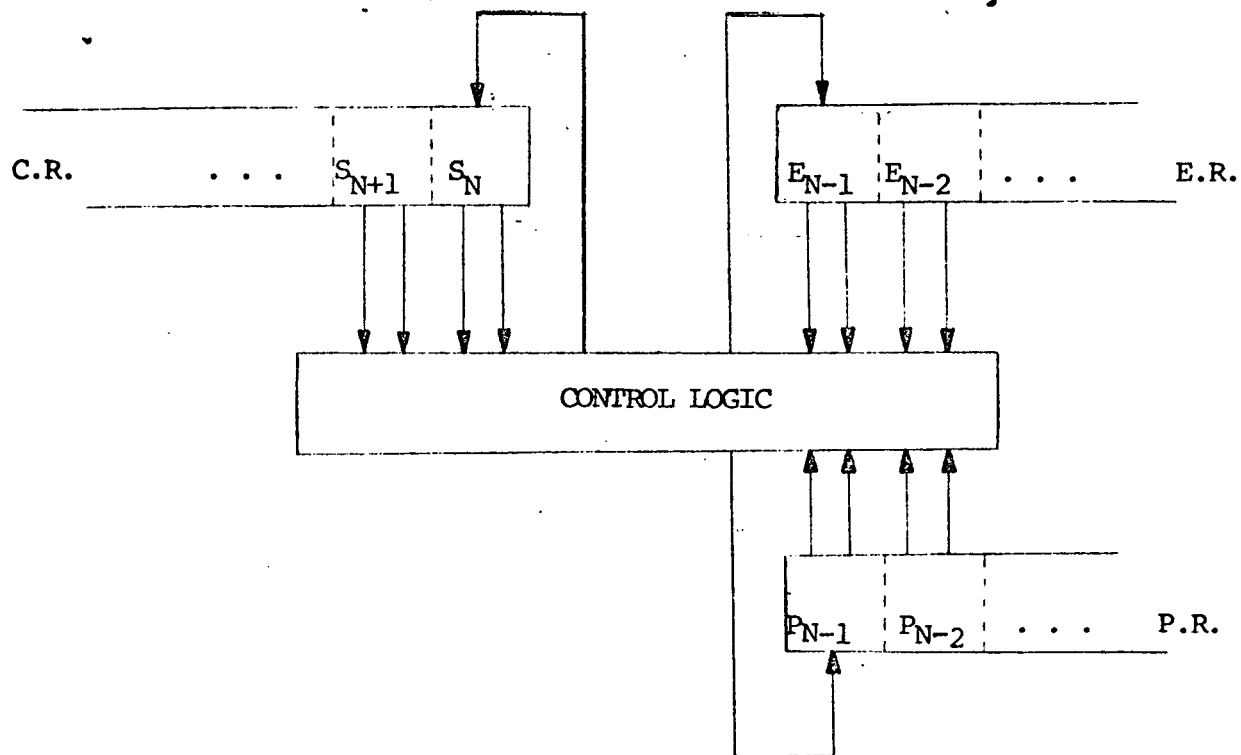


Figure A2-3.4.1.2-1. Relation of Control Logic to Registers

The discussion of the product, Fwd Poly EN, is carried out in much the same fashion, it too must be predicted a period early.

The Fwd and Rev terms are just written down for clock period N. Fwd and Rev are the inverse of each other; so, only an expression for Fwd is given.

Lowering the threshold, according to the flow chart of Section 1, must occur when $L_{N-2}, L_N < T$. In solving the problem of expressing these conditions, consider first the aspect $L_{N-2} < T$.

The test system is designed such that when S_N is present, L_{N-1} (not L_N) is contained in the Metric Counter. Whenever L_{N-1} is zero, L_{N-2} may be less than zero. Two paths lead to $L_{N-1} = 0$; one is a path in which an error has been absorbed, i.e., L_{N-2} was $\Delta - 1$ or $2\Delta - 1$ and the decoder moved through some error condition at level N-2. Moving through the error then dropped L from some value greater than L_{N-1} to zero at L_{N-1} .

The second path to $L_{N-1} = 0$ is the one where L_{N-2} is incremented up by one from $\Delta - 1$ to zero (the Mod- Δ counter rolls over). This is the case where rollover indicates a raising of the threshold and the case where $L_{N-2} < T$.

The condition $L_{N-2} < T$ then will be determined by data from the E and P registers. When L shifted to value $L_{N-1} = 0$, bits E_{N-1} and P_{N-1} were determined. $L_{N-2} < T$ if $E_{N-1} = P_{N-1} = 0$.

The second aspect to be considered in lowering the threshold is the condition $L_N < T$. $L_N < T$ if $L_{N-1} = 0$ and $S_N = 1$. A second case occurs when $L_{N-1} = 0$ and $I_N = 1$. ($L_N < 0$ for this second case since the decoder, in moving forward from this point, must hypothesize one error if $S_N = 1$ or two errors for $S_N = 0$; with $L_{N-1} = 0$, any error causes $L_N < T$.)

These cases are shown in Figure A2-3.4.1.2-2.

Both above aspects neglect approaching the decision block, $L_{N-2}:T$, via the $H = 1$ path. Neglecting this path is required. Recall that $H = 1$ implies the branch then present has had both of its paths searched. This means that the decoder first searched the path corresponding to $I_N = 0$ and, second, proceeded forward on the path corresponding to $I_N = 1$. Proceeding forward under this condition required the Metric absorb the effect of at least one error. Backing into such a branch with $L_{N-1} = 0$ requires that $L_{N-2} > L_{N-1}$.

One last item neglected above was the approach to $T \rightarrow T - \Delta$ via the box N:1. This approach will be referred to by the term Max. Rev.

Summarizing, the above gives:

$$T \rightarrow T - \Delta = (L_{N-1} = 0) \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot \bar{H} \cdot (S_N + I_N) + \text{Max. Rev.}$$

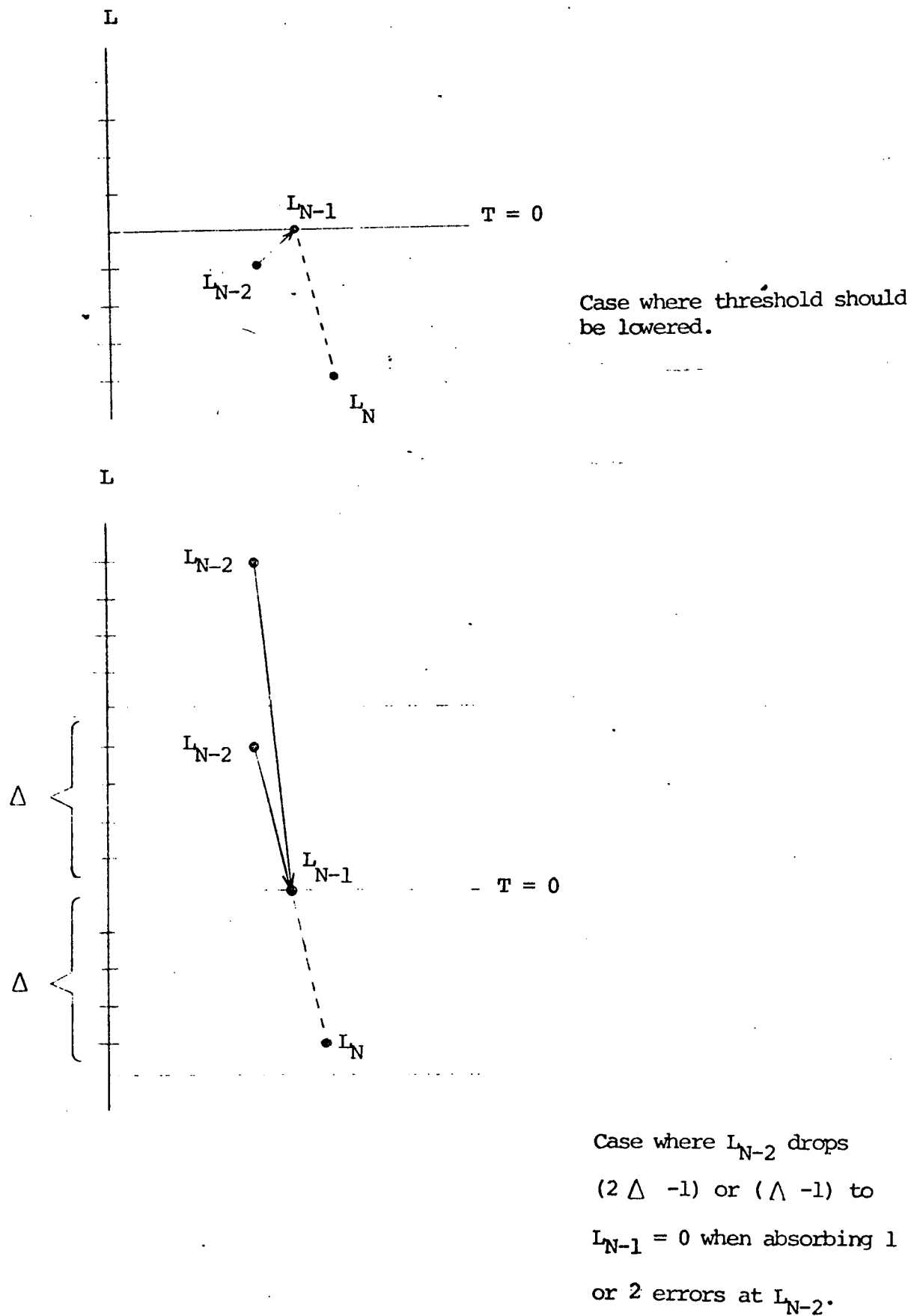


Figure A2-3.4.1.2-2. Threshold Lowering

From the design point of view, it is important to form the product, $T \rightarrow T - \Delta$, early in the clock period. Since the decoder operates at a clock rate of 20 MHz, little time is available for product formation and distribution. In the test system, $T \rightarrow T - \Delta$ is predicted a period early so that it is available from flip-flop at the very beginning of the period requiring its use. Qualification of this flip-flop is discussed below.

One of the terms of $T \rightarrow T - \Delta$ that must be predicted is $L_{N-1} = 0$. The explanation below indicates that L will change to $L = 0$ under these conditions:

$$(L = \Delta - 1) \cdot \text{Fwd} + (L = 1) \cdot \bar{P}_{N-1} \cdot \bar{E}_{N-1} \cdot \text{Rev} = 1$$

In the first case, if $L = (\Delta - 1)$ and the decoder is set to move forward, L will change by one count to $L = 0$. If no error was absorbed, the forward move will imply that the threshold was raised in the process. In any case $(L = \Delta - 1) \cdot \text{Fwd}$ will be followed by $L = 0$. (The case $(L = 2\Delta - 1) \cdot \bar{S}_N \cdot I_N \cdot \text{Fwd}$ is omitted since the threshold will never be lowered immediately following this move.)

In the second case, more than $(L = 1) \cdot \text{Rev}$ is required to predict that $L = 0$ will follow. The decoder could move from $L = 1$ backward to $L = 0$ or $L = \Delta$ or $L = 2\Delta$. Originally when the decoder moved forward to $L = 1$, the move may have been through 1 error, 2 errors or no errors; now that the decoder is moving backward the Metric must be restored to the value it contained prior to the forward move to $L = 1$. To move from $L = 1$ backward to $L = 0$ requires that no errors were absorbed during the forward move to $L = 1$; this is signified by $E_{N-1} = P_{N-1} = 0$.

The product $\bar{E}_{N-1} \cdot \bar{P}_{N-1}$ of the equation for $T \rightarrow T - \Delta$ can be predicted from the earlier equations given for E_{N-1} and P_{N-1} or a table can be constructed as follows:

S_N	I_N	E_{N-2}	P_{N-2}	Fwd	Rev	$T \rightarrow T - \Delta$	E_{N-1}	P_{N-1}
0	0	x	x	1	0	x	0	0
0	x	x	x	1	0	1	0	0
0	1	x	x	1	0	0	1	1
1	0	x	x	1	0	x	1	0
1	1	x	x	1	0	0	0	1
x	x	0	0	0	1	x	0	0

The table just reflects the equations for E_{N-1} and P_{N-1} . From it one can write

$$\begin{aligned}\bar{E}_{N-1} \cdot \bar{P}_{N-1} &= (\bar{S}_N \bar{T}_N + \bar{S}_N (T \rightarrow T - \Delta)) \cdot \text{Fwd} \\ &+ \bar{E}_{N-2} \cdot \bar{P}_{N-2} \cdot \text{Rev.}\end{aligned}$$

$(T \rightarrow T - \Delta)$ does not occur in adjacent clock periods; the pattern is one of lowering T , making at least one move forward, violating the threshold ($L_N < T$) and then backing and checking. So, only terms

$$\bar{S}_N \cdot \bar{T}_N \cdot \text{Fwd} + \bar{E}_{N-2} \cdot \bar{P}_{N-2} \cdot \text{Rev}$$

are of any interest.

Next, consider predicting when $S_N = 1$. In the forward mode, S_N depends on S_{N+1} , the younger syndrome bit, and whether S_{N+1} is to be changed by the Mod-2 addition of the polynomial term, Poly $N+1$.

Recall that when $S_N = 1$, the decoder will try predicting $E_N = 1$ and $P_N = 0$. This guess implies S_N is affecting the younger syndrome bit and, when shifting E_N and P_N into E_{N-1} and P_{N-1} flip-flops, the decoder must remove the effect of S_N by Mod-2 adding the generator polynomial bits to the younger syndrome bits. Hence,

$$S_N = (S_{N+1} \oplus \text{Poly}_{N+1} \cdot \text{Fwd Poly Enable}) \cdot \text{Fwd}$$

In the reverse mode S_N is just the Mod-2 sum of E_{N-1} and P_{N-1} :

$$S_N = (E_{N-1} \oplus P_{N-1}) \cdot \text{Rev.}$$

At this point, consider consolidating the various terms. Requirements for predicting \bar{H} and I_N are then easily added.

The best approach is to group terms into ones used in the forward mode and those used in reverse.

$$[(L_{N-1} = 0) \cdot \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot S_N]_{\text{Fwd}} \text{ is predicted by}$$

$$(L = \Delta - 1) \cdot \bar{S}_N \bar{T}_N \cdot (S_{N+1} \oplus \text{Poly}_{N+1} \cdot \text{Fwd Poly EN}) \cdot \text{Fwd.}$$

$$[(L_{N-1} = 0) \cdot \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot S_N]_{\text{Rev}}, \text{ predicted by}$$

$$[(L = 1) \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot \bar{E}_{N-2} \cdot \bar{P}_{N-2} \cdot (E_{N-1} \oplus P_{N-1}) \cdot \text{Rev},$$

never happens.

So far, the expression predicting $(L_{N-1} = 0) \cdot \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot S_N$ has been found. Next, consider predicting $(L_{N-1} = 0) \cdot \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot I_N$.

I_N will be set whenever the decoder makes a reverse shift. So, $(L_{N-1} = 0) \cdot \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot I_N$ will be predicted by the reverse terms of $(L_{N-1} = 0) \cdot \bar{E}_{N-1} \cdot \bar{P}_{N-1}$. These are

$$[(L = 1) \cdot \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot \bar{E}_{N-2} \cdot \bar{P}_{N-2}] \cdot \text{Rev}$$

Finally, the two terms obtained thus far must be ANDed with the terms predicting \bar{H} . H will be predicted when $P_{N-1} \cdot \text{Rev}$ exists. $P_{N-1} = 1$ indicates a forward move has been made with $L_{N-1} = 1$. ANDing P_{N-1} with Rev then indicates the decoder is about to back into a branch level where both paths have been tried. \bar{H} then will be predicted by

$$\text{Fwd} + \bar{P}_{N-1} \cdot \text{Rev} = 1.$$

Predicting \bar{H} , therefore, adds nothing to the two terms already obtained. The flip-flop, $T \rightarrow T - \Delta$, then is qualified by

$$\begin{aligned} & [(L = \Delta - 1) \cdot \bar{S}_N \bar{T}_N \cdot (S_{N+1} \oplus \text{Poly}_{N+1} \cdot \text{Fwd Poly EN}) \cdot \text{Fwd} \\ & + [(L = 1) \cdot \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot \bar{E}_{N-2} \cdot \bar{P}_{N-2}] \cdot \text{Rev} \\ & + \text{Max. Rev} = 1. \end{aligned}$$

At this point it seems appropriate to describe the details involved in controlling addition of the generator polynomial. The problem has two basic aspects. First, on forward moves where E_{N-1} is to be set to a one, the addition is required; this addition is performed to remove effect of S_N from younger syndromes. Second, on reverse moves where $E_{N-1} (=1)$ is about to be combined with P_{N-1} to reform a syndrome bit, the addition is required; this addition is performed to cancel an earlier addition.

The last aspect depends only on E_{N-1} . If $E_{N-1} = 1$ and a reverse shift occurs, an addition must take place.

The first aspect is more complex. Addition should occur if $E_{N-1} = 1$ is hypothesized and a forward move takes place. Recall from the expression for E_{N-1} that E_{N-1} will be set on forward when

$$(S_N \oplus I_N) \cdot (T \rightarrow T - \Delta) \cdot \text{Fwd} + S_N \cdot (T \rightarrow T - \Delta) \cdot \text{Fwd} = 1.$$

This expression should qualify the polynomial addition. Now since the product is needed early in the clock period, a flip-flop output called Fwd Poly EN is ANDed with Fwd and used as the qualifying signal. The prediction of the expression,

$$(S_N \oplus I_N) \cdot (\overline{T \rightarrow T - \Delta}) + S_N \cdot (T \rightarrow T - \Delta),$$

is used to qualify the Fwd Poly EN flip-flop; just how this is done is described below.

Suppose an addition on a forward move is called for; what this implies depends on the move the decoder made in the clock period just prior to the one containing the addition.

If the decoder moved forward during the preceding clock period, $I_N = 0$ and the expression

$$(S_N \oplus I_N) \cdot (\overline{T \rightarrow T - \Delta}) + S_N \cdot (T \rightarrow T - \Delta)$$

becomes simply S_N .

If the decoder moved backward during the preceding clock period, $I_N = 1$ and expression above becomes

$$S_N \oplus (\overline{T \rightarrow T - \Delta}).$$

So, qualifying Fwd Poly EN amounts to predicting S_N during forward moves and $S_N \oplus (\overline{T \rightarrow T - \Delta})$ during reverse moves.

Predicting $S_N = 1$ was already done for flip-flop $(T \rightarrow T - \Delta)$. The expression used is:

$$[S_{N+1} \oplus \text{Poly}_{N+1} \cdot \text{Fwd Poly EN}] \cdot \text{Fwd} = 1.$$

Predicting $S_N \oplus (\overline{T \rightarrow T - \Delta})$ can be done using two results from earlier discussions.

First recall that on reverse shifts, $T \rightarrow T - \Delta$ is predicted by

$$[(L = 1) \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot \bar{E}_{N-2} \cdot \bar{P}_{N-2}] \cdot \text{Rev} + \text{Max Rev.} = 1.$$

Recall also that on reverse shifts $S_N = 1$ is predicted by

$$(E_{N-1} \oplus P_{N-1}) \cdot \text{Rev} = 1.$$

Combining the above, $S_N \oplus (\overline{T \rightarrow T - \Delta})$ can be predicted by the terms

$$(E_{N-1} \oplus P_{N-1}) \cdot [(L = 1) \cdot \overline{E}_{N-1} \cdot \overline{P}_{N-1} \cdot \overline{E}_{N-2} \cdot \overline{P}_{N-2}] \cdot \text{Rev} +$$

$$(E_{N-1} \oplus P_{N-1}) \cdot (\text{Max Rev}) \cdot \text{Rev} +$$

$$\overline{(E_{N-1} \oplus P_{N-1})} \cdot \overline{[(L = 1) \cdot \overline{E}_{N-1} \cdot \overline{P}_{N-1} \cdot \overline{E}_{N-2} \cdot \overline{P}_{N-2}]} \cdot \overline{(\text{Max Rev})} \cdot \text{Rev} = 1.$$

ORing the forward and reverse terms obtained then gives the final expression required to qualify the Fwd Poly EN flip-flop.

Next, consider the most basic of signals required in the decoder, Fwd and Rev.

The decoder moves each clock period; Fwd and Rev are just the inverse of each other.

When a forward move is to be made, $L_N \geq T$. In the decoder L_{N-1} is actually used in the test. L_N will be greater than or equal to T if

$$(L_{N-1} \geq 2\Delta - 1) + (L_{N-1} \geq \Delta - 1) \cdot S_N + \overline{S}_N \overline{T}_N = 1$$

In the decoder, the Δ counter has stages $\Delta 2^0$ through $\Delta 2^3$. The Mod Δ counter has stages $M\Delta 2^0$ through $M\Delta 2^2$. A special flip-flop makes available a signal, $2^0/2^0$. When $\Delta = 5$, $2^0/2^0 = M\Delta 2^0$; when $\Delta = 6$, $2^0/2^0 = M\Delta 2^0$. Using these symbols in the equation above gives

$$(\Delta 2^3 + \Delta 2^2 + \Delta 2^1 + \Delta 2^0 \cdot M\Delta 2^2 \cdot 2^0/2^0) + (\Delta 2^0 + M\Delta 2 \cdot 2^0/2^0) \cdot S_N + \overline{S}_N \overline{T}_N = 1.$$

The expression above gives $L_N \geq T$. Fwd requires ANDing $L_N \geq T$ with \overline{H} ; $H = 1$ implies the decoder should skip the then present branch and back up one step further.

The final expression can now be written as

$$\Delta 2^3 \cdot \overline{H} + \Delta 2^2 \cdot \overline{H} + \Delta 2^1 \cdot \overline{H} + \Delta 2^0 \cdot M\Delta 2^2 \cdot 2^0/2^0 \cdot \overline{H} + \Delta 2^0 \cdot S_N \cdot \overline{H} + M\Delta 2^2 \cdot 2^0/2^0 \cdot S_N \cdot \overline{H} + \overline{S}_N \cdot \overline{T}_N \cdot \overline{H} = \text{Fwd}$$

Instead of generating these signals during the period they are used, Fwd and Rev can be predicted a clock period earlier and appear at the output of a flip-flop at the beginning of the clock period they are needed. This eliminates a delay of an OR circuit used to generate the expression above.

To obtain an expression used to qualify the Fwd/Rev flip-flop, consider the possible transitions the decoder may make to a forward state. These are tabulated below. For brevity let

$$S_p = S_{N+1} \oplus \text{Poly}_{N+1} \cdot \text{Fwd Poly } E_N;$$

S_p will represent the predicted syndrome bit wherever it is used in the following.

Possible Beginning State	Possible State Predicted
$\bar{S}_N \cdot \bar{T}_N \cdot \text{Fwd}$	$\bar{S}_p \cdot \text{Fwd}$
$\bar{S}_N \cdot \bar{T}_N \cdot \text{Fwd}$	$S_p \cdot \text{Fwd}$
$\bar{S}_N \cdot I_N \cdot \text{Fwd}$	$\bar{S}_p \cdot \text{Fwd}$
$\bar{S}_N \cdot I_N \cdot \text{Fwd}$	$S_p \cdot \text{Fwd}$
$S_N \cdot \bar{T}_N \cdot \text{Fwd}$	$\bar{S}_p \cdot \text{Fwd}$
$S_N \cdot \bar{T}_N \cdot \text{Fwd}$	$S_p \cdot \text{Fwd}$
$S_N \cdot \bar{T}_N \cdot \text{Fwd}$	$\bar{S}_p \cdot \text{Fwd}$
$S_N \cdot \bar{T}_N \cdot \text{Fwd}$	$S_p \cdot \text{Fwd}$
$E_{N-1} \cdot \bar{P}_{N-1} \cdot \text{Rev}$	$\bar{E}_N \cdot P_N \cdot \text{Fwd}$
$\bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot \text{Rev}$	$E_N \cdot P_N \cdot \text{Fwd}$

Note that transitions from states $\bar{E}_{N-1} \cdot P_{N-1} \cdot \text{Rev}$ and $E_{N-1} \cdot P_{N-1} \cdot \text{Rev}$ are accompanied by an increase in the Metric and by the setting of H to a 1. Hence, except when Max. Rev is reached, no threshold lowering can occur and a second reverse move must follow the presently scheduled reverse move.

Of all the possible transitions listed above, five dictate a forward move by the decoder without immediate reference to the metric. Combining beginning and predicted states, these five transitions are represented by:

$$\begin{aligned}
& \bar{S}_N \cdot \bar{I}_N \cdot \bar{S}_P \cdot \text{Fwd} \\
& \bar{S}_N \cdot I_N \cdot \bar{S}_P \cdot \text{Fwd} \\
& S_N \cdot \bar{I}_N \cdot \bar{S}_P \cdot \text{Fwd} \\
& S_N \cdot I_N \cdot \bar{S}_P \cdot \text{Fwd} \\
& E_{N-1} \cdot \bar{P}_{N-1} \cdot \text{Rev}
\end{aligned}$$

The term Fwd in the first four combinations implies the metric will remain positive for the given state of S_N and I_N . The term \bar{S}_P says $\bar{S}_N \cdot \bar{I}_N = 1$ for the state to which the decoder is moving. So, the product, $\bar{S}_P \cdot \text{Fwd} = 1$, in each of the four says a forward move should follow the presently scheduled forward move.

The fifth combination will always be followed by a forward move with $\bar{E}_N \cdot P_N = 1$.

The remaining five transitions require certain minimum metric values before the forward state they predict can occur. Two of the five require the same metric conditions and can be combined. These are the transitions beginning with the state $S_N \cdot \bar{I}_N \cdot \text{Fwd}$ or $S_N \cdot I_N \cdot \text{Fwd}$ and ending with state $S_P \cdot \text{Fwd}$. For these two cases the decoder must have as a metric value:

$$\Delta 2^3 + \Delta 2^2 + \Delta 2^1 + \Delta 2^0 \cdot M\Delta 2^2 + \Delta 2^0 \cdot M\Delta 2^1 \cdot M\Delta 2^0 \cdot (\Delta = 5) = 1$$

The metric counter's most significant stages, $\Delta 2^3$ and $\Delta 2^2$, are large enough to absorb any of the errors that may occur for the transitions listed above. Hence, if predicted $H = 0$, the presence of either $\Delta 2^3$ or $\Delta 2^2$ is sufficient to predict a forward move following the presently scheduled move - be it forward or reverse. With this idea in mind, $\Delta 2^3 + \Delta 2^2$ can be left out of the expression given in the paragraph above and all other expressions where a minimum metric value is required and included as a separate term in the overall expression used to qualify the Fwd/Rev flip-flop.

Considering the ideas suggested so far, the Fwd/Rev flip-flop should be set when

$$\begin{aligned}
& \bar{S}_P \cdot \text{Fwd} + E_{N-1} \cdot \bar{P}_{N-1} \cdot \text{Rev} + \bar{S}_N \cdot \bar{I}_N \cdot S_P \cdot (\Delta 2^1 + \Delta 2^0 + M\Delta 2^2 + \\
& (\Delta = 5) \cdot M\Delta 2^1 \cdot M\Delta 2^0) + S_N \cdot S_P \cdot (\Delta 2^1 + \Delta 2^0 \cdot M\Delta 2^2 + \Delta 2^0 \cdot M\Delta 2^1 \cdot \\
& M\Delta 2^0 (\Delta = 5)) \cdot \text{Fwd} + \bar{S}_N \cdot I_N \cdot S_P \cdot (\Delta 2^1 \cdot \Delta 2^0 + \Delta 2^1 \cdot M\Delta 2^2 + \Delta 2^1 \cdot \\
& M\Delta 2^1 \cdot M\Delta 2^0 \cdot (\Delta = 5)) \cdot \text{Fwd} + \bar{E}_{N-1} \cdot \bar{P}_{N-1} \cdot \Delta 2^1 \cdot \text{Rev} + (\Delta 2^3 + \Delta 2^2) \cdot \\
& \bar{H}_P = 1
\end{aligned}$$

where $\bar{H}_p = \overline{P_{N-1} \cdot \text{Rev}}$.

The above expression does not reflect the need of going forward when the threshold is lowered. Note, too, that when the flip-flop generating $T \rightarrow T - \Delta$ is set, the metric value used in qualifying the Fwd/Rev flip-flop will not include the effect of the lowering. In the test set a separate flip-flop is set coincidentally with the " $T \rightarrow T - \Delta$ " flip-flop, or can be set when qualified by $(T \rightarrow T - \Delta) \cdot (\bar{S}_p + S_N)$; this separate flip-flop includes the effects of threshold lowering and is used to preset the Fwd/Ref flip-flop in the Fwd state for these cases.

• The Fano control logic discussed thus far is implemented on drawing 528760, sheets 1 through 4. Signals $(+\Delta - 1)$, $(+2\Delta - 1)$ and $(-2\Delta + 1)$ which appear on the drawings are discussed in the overall explanation of the metric counter design which follows.

In the discussions above, much has been implied about the Metric counter; these implications are integrated below in the description of the counter's overall design.

Most basically, the Metric counter, made up of the Mod- Δ and Δ counters, is an up-down counter. Provisions are made to increment it by $+1$, $+\Delta - 1$ and $+2\Delta - 1$; it can be decremented by -1 , $-\Delta + 1$ and $-2\Delta + 1$.

The Mod- Δ portion of the Metric can only change by ± 1 . So, only Fwd and Rev are needed as qualifiers.

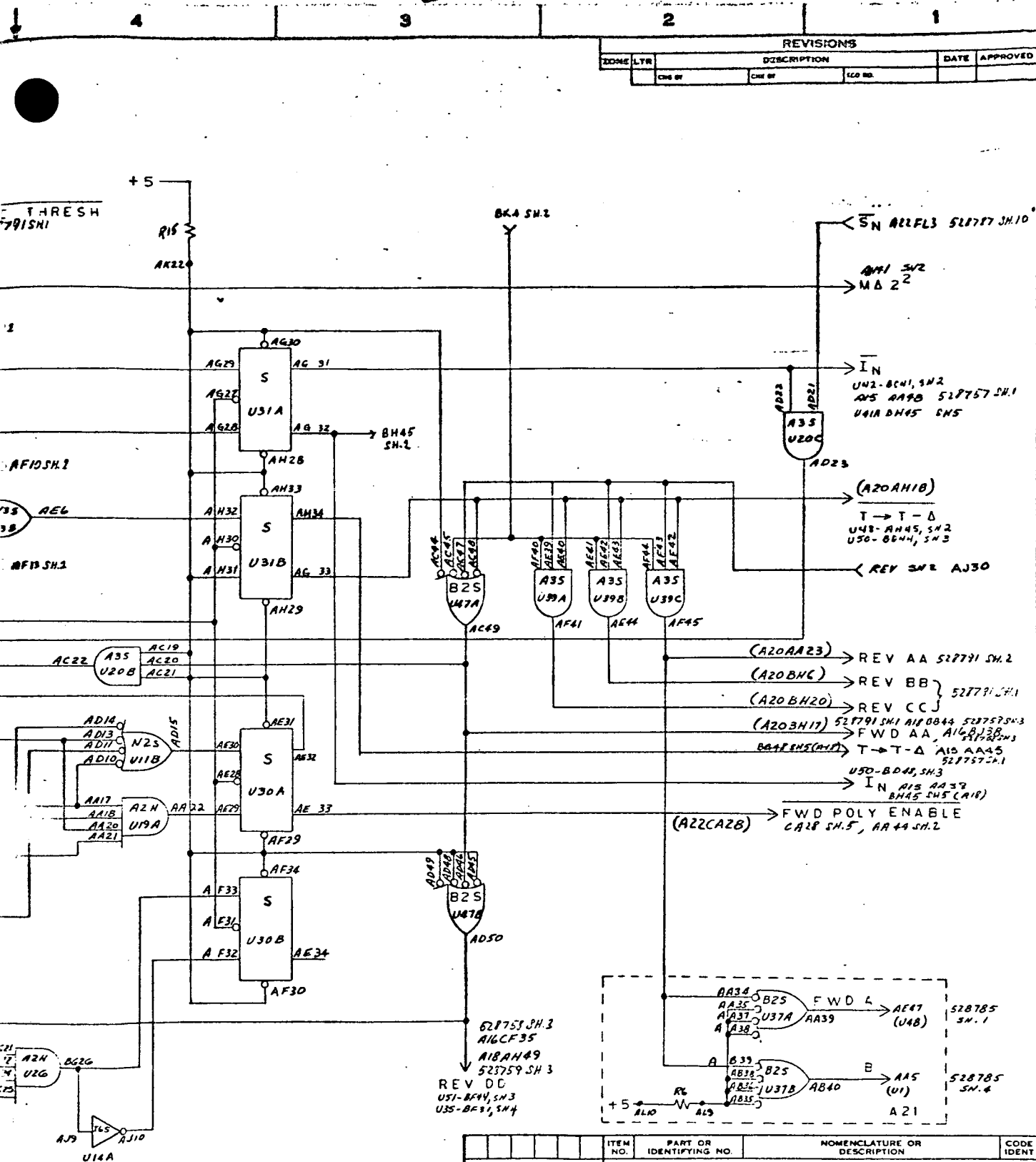
The Δ portion of the Metric counter can be changed by $\pm\Delta$ or $\pm 2\Delta$. Increases of $+\Delta$ or $+2\Delta$ in the Δ counter are always accompanied by a decrease of 1 in the Mod- Δ counter, thus, causing the overall Metric value to change by $+\Delta - 1$ or $+2\Delta - 1$. Decreases of $-\Delta$ or -2Δ in the Δ counter are similarly always accompanied by an increase of 1 in the Mod- Δ counter, thus, causing the overall Metric value to change by $-\Delta + 1$ or $-2\Delta + 1$.

Changes to be made in the Metric Value are indicated by the following lines: $(+2\Delta - 1)$, $(+\Delta - 1)$, $(-2\Delta + 1)$ and $(-\Delta + 1)$. These are conditional and their effect depends on Fwd and Rev levels and on carries or borrows generated by the Mod- Δ counter.

In general, the Metric control lines indicate certain decoder conditions.

1. $(+2\Delta - 1)$ indicates $E_{N-1} = P_{N-1} = 1$; in the Δ counter $(+2\Delta - 1)$ is ANDed with Rev.; should a reverse move occur, the Metric will be credited with a $+2\Delta - 1$ lost during an earlier forward move.
2. $(+\Delta - 1)$ indicates $E_{N-1} \oplus P_{N-1} = 1$; here as above the signal is ANDed with Rev.; should a reverse move occur, the metric will be credited with a $+\Delta - 1$ lost during an earlier forward move.

EOLDOUT FRAME 2



						ITEM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	CODE IDENT
QUANTITY REQUIRED						LIST OF MATERIALS OR PARTS LIST			

PART TO BE FREE OF BURRS AND SHARP EDGES
 COMMERCIAL TOLERANCES APPLY TO STOCK SIZES
 MANUFACTURING TOLERANCES PER 900002
 WORMMESH PER 900050
 SCREW THREADS PER MIL-STD-8
 APPROPRIATIONS PER MIL-STD-12
 LOGIC SYMBOLS PER MIL-STD-406
 DIMENSIONING AND TOLERANCING PER USAS
 314.5-66
 ELECTRICAL AND ELECTRONIC DIAGRAM PER USAS
 314.5-66
 GRAPHIC SYMBOLS FOR ELECTRICAL AND
 ELECTRIC DIAGRAM PER USAS 312.4-67
 ELECTRICAL AND ELECTRONIC REFERENCE
 DESIGNATIONS PER USAS 72.12-65
 WELDING SYMBOLS PER A5.22-68

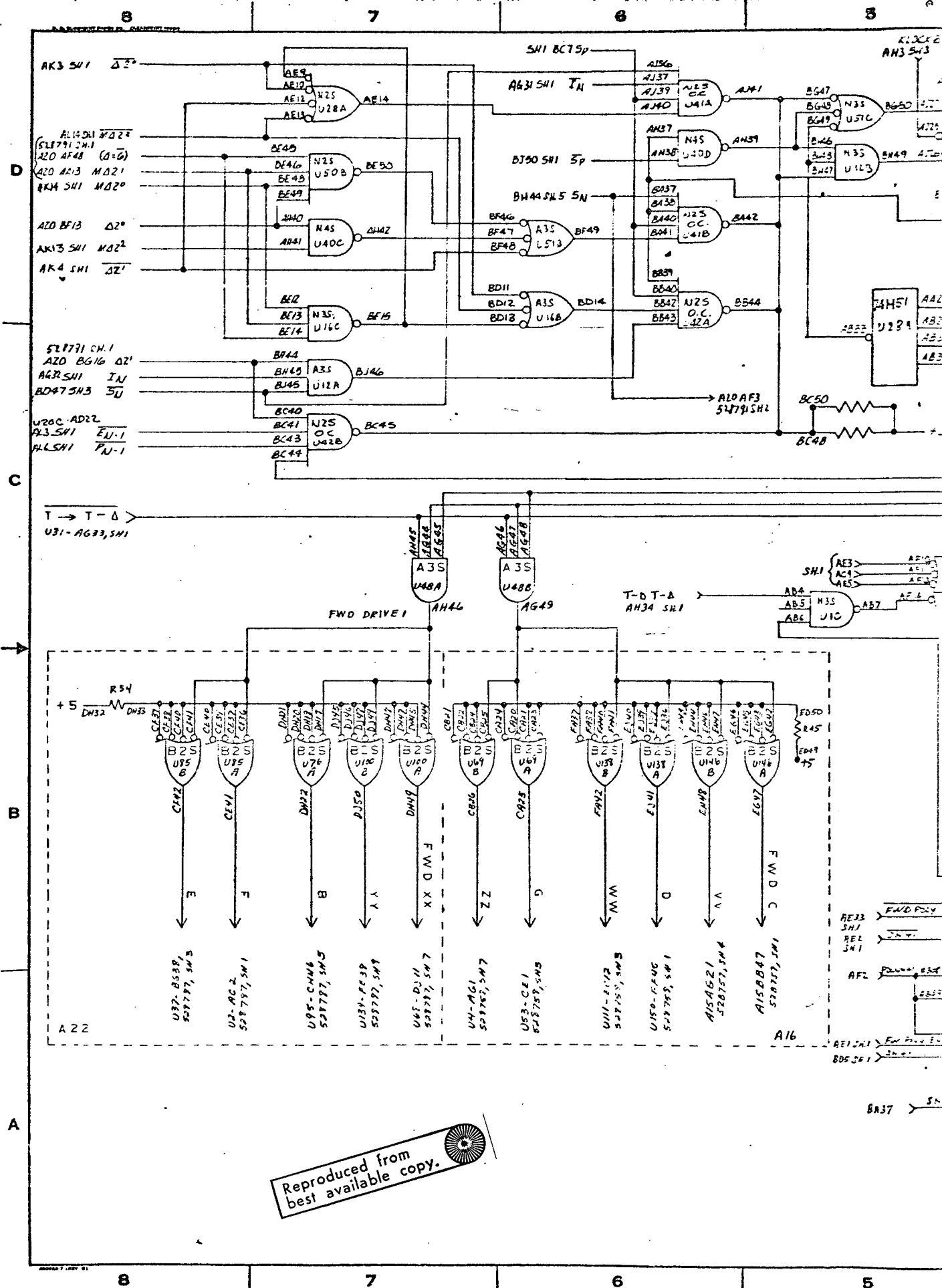
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
AND INCLUDE APPLIED FINISH
TOLERANCES

3 PLACE DIM	3 PLACE DIM	ANGLE
"	"	"

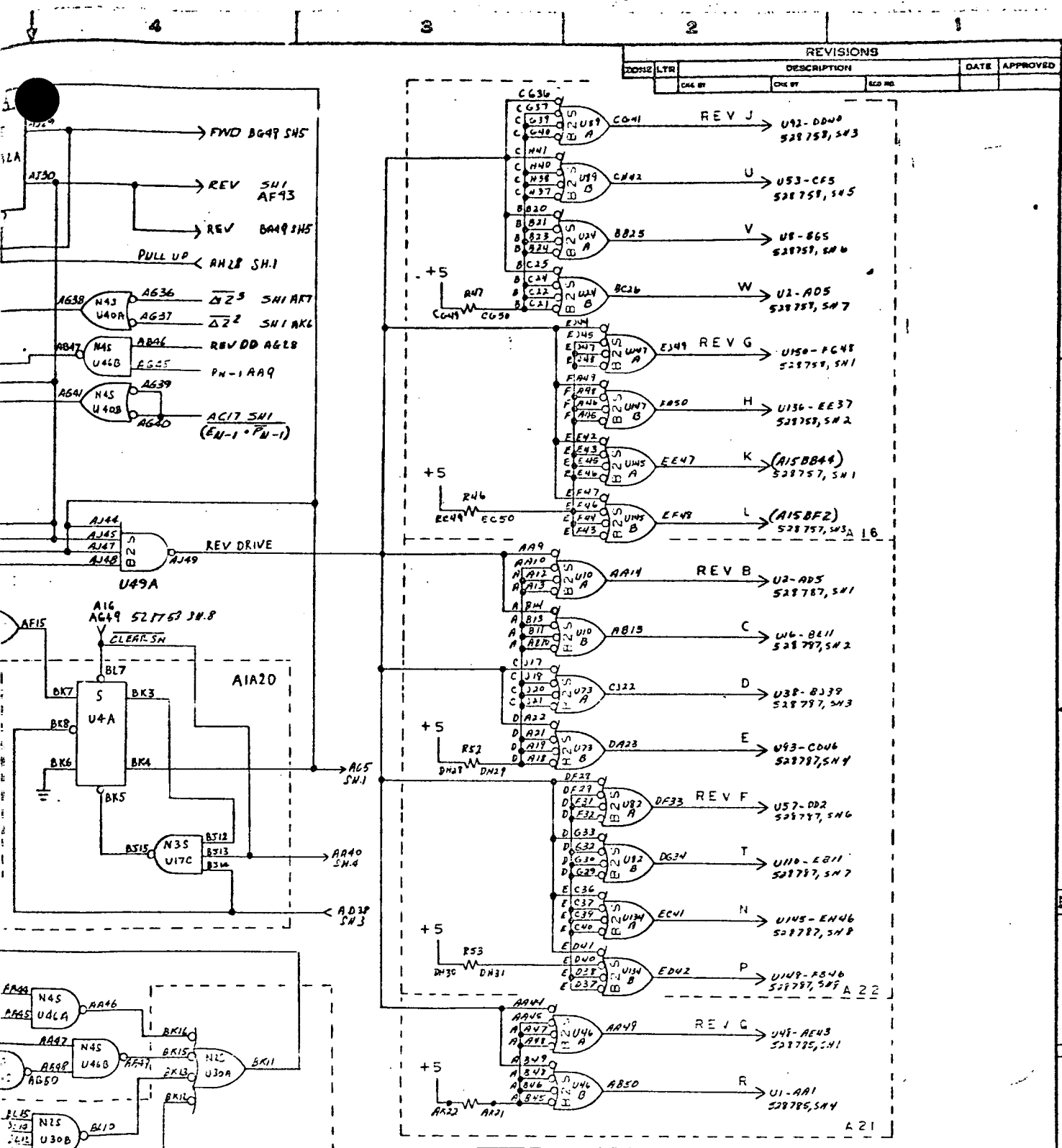
CONTRACT NO	
OR BY	CNR BY
ENGINEER	
MALNAR	
PROJECT ENGINEER	
L.S. down 11/29/77	
APPROVAL	
APPROVAL	

RADIATION INCORPORATED SUBSIDIARY OF HARRIS-INTERTYPE CORPORATION, MELBOURNE, FLORIDA			
TITLE FANC DECODER (CONTROL LOGIC)			
SIZE D	COOF IDENT NO 91417	528760	REV 0

FOLDOUT FRAME



EOLDOUT FRAME 2



REVISIONS				DATE	APPROVED
DATE	DESCRIPTION	CHK BY	CHK BY	ECG NO.	

					ITEM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	CODE IDENT		
QUANTITY REQUIRED					LIST OF MATERIALS OR PARTS LIST					
TWICE SPECIFIED ARE IN INCHES APPLIED FINISH RANGES PLACE DIM. ANGLES C					CONTRACT NO.					
					DR BY		CHK BY		RADIATION INCORPORATED SUBSIDIARY OF HARRIS-INTERTYPE CORPORATION, MELBOURNE, FLORIDA	
					ENGINEER					
					MALNAR					
					PROJECT ENGINEER					
					J. J. J. 11/29/71					
TITLE				FANO DECODER (FND AND REV DISTRIBUTION)						
APPROVAL										
APPROVAL										
SIZE		CODE IDENT NO				REV				
D		91417				528760				
SCALE					SHEET 2 OF 5					

PART TO BE FREE OF BURRS AND SHARP EDGES
COMMERCIAL TOLERANCES APPLY TO STOCK SIZES
MANUFACTURING TOLERANCES PER 900002
WORKMANSHIP PER 900002
SCREEN THICKNESS PER MIL STD-8
ABBREVIATIONS PER MIL STD-12
LOGIC SYMBOLS PER MIL STD-808
DIMENSIONING AND TOLERANCING PER USAS
Y14.5-66
ELECTRICAL AND ELECTRONIC DIAGRAM PER USAS
Y11.15-66
GRAPHIC SYMBOLS FOR ELECTRICAL AND
ELECTRONIC DIAGRAM PER USAS Y32.2-67
ELECTRICAL AND ELECTRONIC REFERENCE
SYMBOLS PER USAS Y32.16-65
WELDING SYMBOLS PER AWS A2.4-58

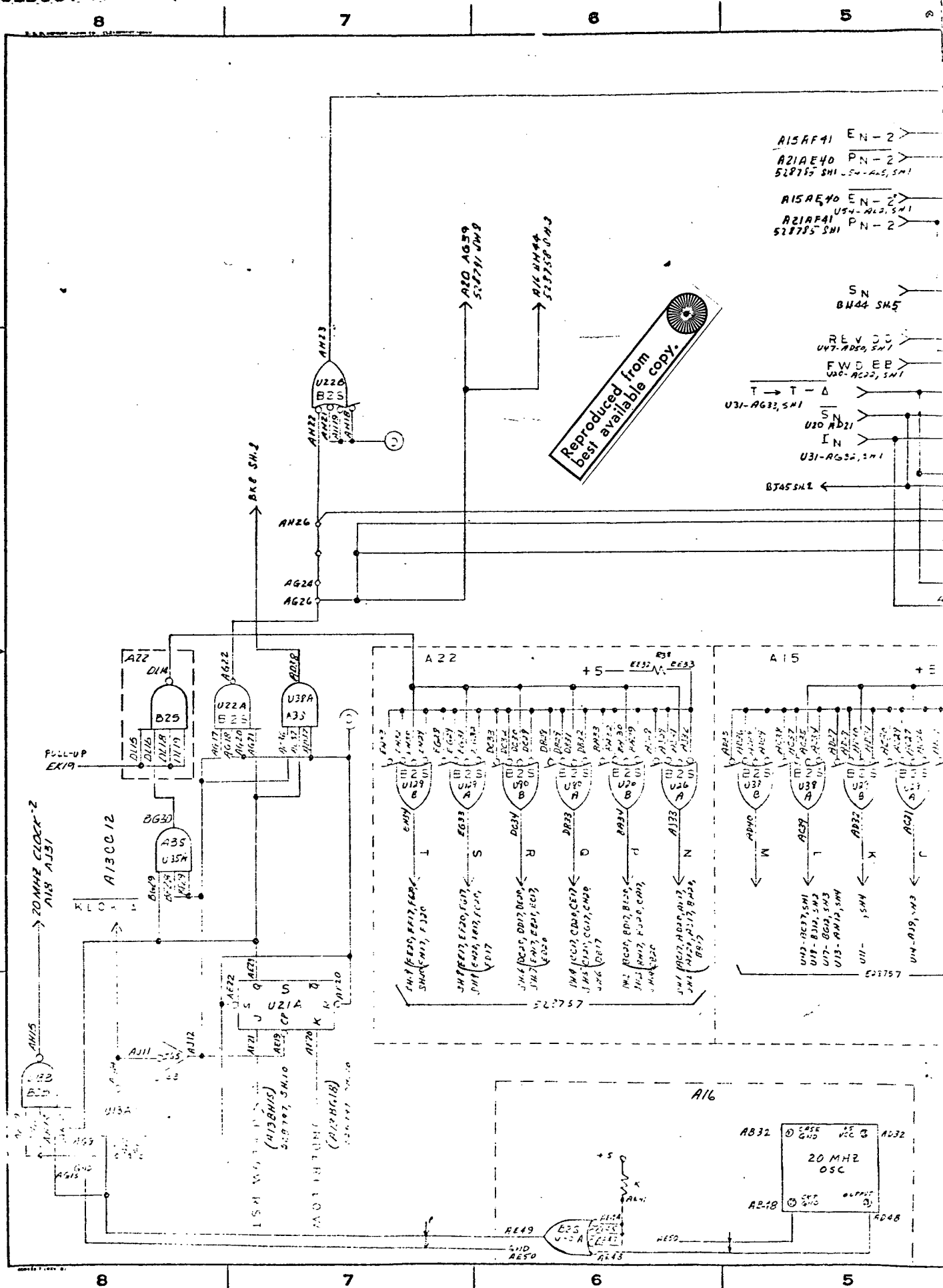
APPLICATION	NEXT ASBY	USED ON

D

C

B

A



A15AF41 EN-2
A21AE40 PN-2
518755 SHI 50-A25, SHI
A15AE40 EN-2
A21AF41 PN-2
518755 SHI 50-A25, SHI

SN
B444 SH5

REV 00
U17-A250, SHI

FWD EEP
U10-A222, SHI

T → T-Δ
U31-A632, SHI

SN
U20 AD21

I N
U31-A632, SHI

BT45 SH1

A 22

+5

A 15

+5

A 16

+5

A 17

+5

A 18

+5

A 19

+5

A 20

+5

A 21

+5

A 22

+5

A 23

+5

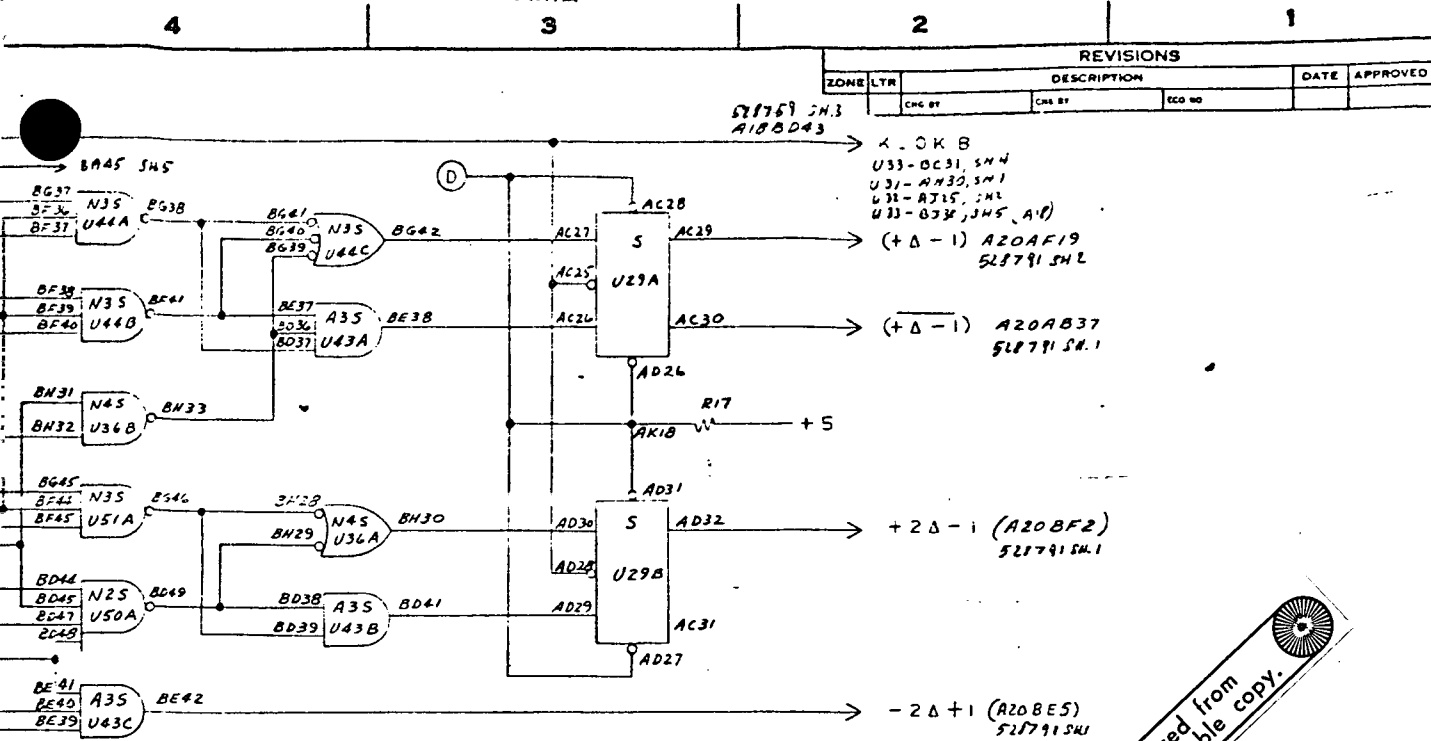
A 24

+5

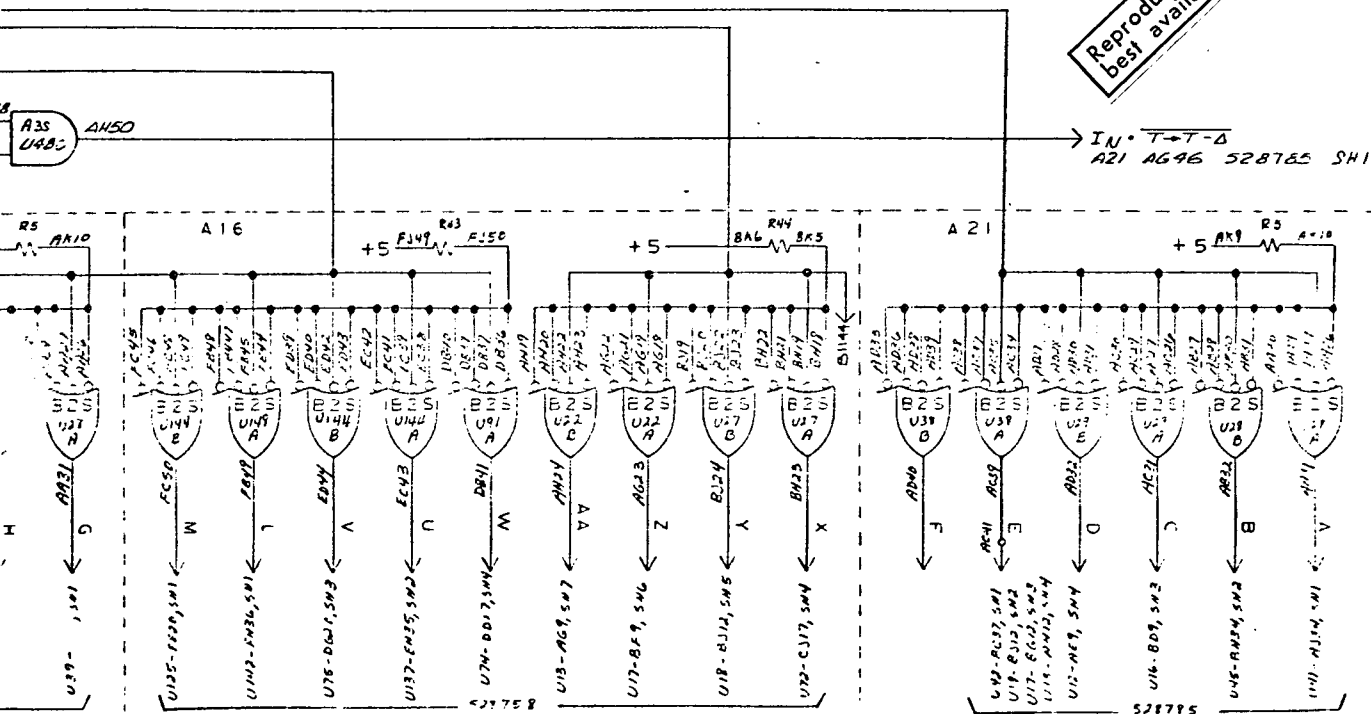
A 25

+5

FOLDOUT FRAME 2



5)
SRI



					ITEM NO	PART OR IDENTIFYING NO	NOMENCLATURE OR DESCRIPTION	CODE IDENT
QUANTITY REQUIRED					LIST OF MATERIALS OR PARTS LIST			

PART TO BE FREE OF BURRS AND SHARP EDGES
COMMERCIAL TOLERANCES APPLY TO STOCK SIZES
MANUFACTURING TOLERANCES PER Y50062
WORKMANSHIP PER Y50060
SCREW THREADS PER MIL STD 9
ABBREVIATIONS PER MIL STD 12
LOGIC SYMBOLS PER MIL STD 806
DIMENSIONING AND TOLERANCING PER USAS
714 5-6
ELECTRICAL AND ELECTRONIC DIAGRAM PER USAS
714 15-6
GRAPHIC SYMBOLS FOR ELECTRICAL AND
ELECTRONIC DIAGRAM PER USAS 722 2-47
ELECTRICAL AND ELECTRONIC REFERENCE
DESIGNATIONS PER USAS 732 16-43
WELDING SYMBOLS PER AWS 42-58

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
AND INCLUDE APPLIED FINISH
TOLERANCES

1. NAME	2. NAME	3. NAME
4. NAME	5. NAME	6. NAME
7. NAME	8. NAME	9. NAME
10. NAME	11. NAME	12. NAME
13. NAME	14. NAME	15. NAME
16. NAME	17. NAME	18. NAME
19. NAME	20. NAME	21. NAME
22. NAME	23. NAME	24. NAME
25. NAME	26. NAME	27. NAME
28. NAME	29. NAME	30. NAME
31. NAME	32. NAME	33. NAME
34. NAME	35. NAME	36. NAME
37. NAME	38. NAME	39. NAME
40. NAME	41. NAME	42. NAME
43. NAME	44. NAME	45. NAME
46. NAME	47. NAME	48. NAME
49. NAME	50. NAME	51. NAME
52. NAME	53. NAME	54. NAME
55. NAME	56. NAME	57. NAME
58. NAME	59. NAME	60. NAME
61. NAME	62. NAME	63. NAME
64. NAME	65. NAME	66. NAME
67. NAME	68. NAME	69. NAME
70. NAME	71. NAME	72. NAME
73. NAME	74. NAME	75. NAME
76. NAME	77. NAME	78. NAME
79. NAME	80. NAME	81. NAME
82. NAME	83. NAME	84. NAME
85. NAME	86. NAME	87. NAME
88. NAME	89. NAME	90. NAME
91. NAME	92. NAME	93. NAME
94. NAME	95. NAME	96. NAME
97. NAME	98. NAME	99. NAME
100. NAME	101. NAME	102. NAME
103. NAME	104. NAME	105. NAME
106. NAME	107. NAME	108. NAME
109. NAME	110. NAME	111. NAME
112. NAME	113. NAME	114. NAME
115. NAME	116. NAME	117. NAME
118. NAME	119. NAME	120. NAME
121. NAME	122. NAME	123. NAME
124. NAME	125. NAME	126. NAME
127. NAME	128. NAME	129. NAME
130. NAME	131. NAME	132. NAME
133. NAME	134. NAME	135. NAME
136. NAME	137. NAME	138. NAME
139. NAME	140. NAME	141. NAME
142. NAME	143. NAME	144. NAME
145. NAME	146. NAME	147. NAME
148. NAME	149. NAME	150. NAME
151. NAME	152. NAME	153. NAME
154. NAME	155. NAME	156. NAME
157. NAME	158. NAME	159. NAME
160. NAME	161. NAME	162. NAME
163. NAME	164. NAME	165. NAME
166. NAME	167. NAME	168. NAME
169. NAME	170. NAME	171. NAME
172. NAME	173. NAME	174. NAME
175. NAME	176. NAME	177. NAME
178. NAME	179. NAME	180. NAME
181. NAME	182. NAME	183. NAME
184. NAME	185. NAME	186. NAME
187. NAME	188. NAME	189. NAME
190. NAME	191. NAME	192. NAME
193. NAME	194. NAME	195. NAME
196. NAME	197. NAME	198. NAME
199. NAME	200. NAME	201. NAME
202. NAME	203. NAME	204. NAME
205. NAME	206. NAME	207. NAME
208. NAME	209. NAME	210. NAME
211. NAME	212. NAME	213. NAME
214. NAME	215. NAME	216. NAME
217. NAME	218. NAME	219. NAME
220. NAME	221. NAME	222. NAME
223. NAME	224. NAME	225. NAME
226. NAME	227. NAME	228. NAME
229. NAME	230. NAME	231. NAME
232. NAME	233. NAME	234. NAME
235. NAME	236. NAME	237. NAME
238. NAME	239. NAME	240. NAME
241. NAME	242. NAME	243. NAME
244. NAME	245. NAME	246. NAME
247. NAME	248. NAME	249. NAME
250. NAME	251. NAME	252. NAME
253. NAME	254. NAME	255. NAME
256. NAME	257. NAME	258. NAME
259. NAME	260. NAME	261. NAME
262. NAME	263. NAME	264. NAME
265. NAME	266. NAME	267. NAME
268. NAME	269. NAME	270. NAME
271. NAME	272. NAME	273. NAME
274. NAME	275. NAME	276. NAME
277. NAME	278. NAME	279. NAME
280. NAME	281. NAME	282. NAME
283. NAME	284. NAME	285. NAME
286. NAME	287. NAME	288. NAME
289. NAME	290. NAME	291. NAME
292. NAME	293. NAME	294. NAME
295. NAME	296. NAME	297. NAME
298. NAME	299. NAME	300. NAME
301. NAME	302. NAME	303. NAME
304. NAME	305. NAME	306. NAME
307. NAME	308. NAME	309. NAME
310. NAME	311. NAME	312. NAME
313. NAME	314. NAME	315. NAME

1. *Journal of Management Studies*, 1990, 27, 1, 1-14.

CONTRACT NO

[illegible]

Σ μ_i	Cost μ_i
-----------	--------------

ENGINEER

92-325

PROJECT ENGINEER
C. A. [Signature] 11/29/71

11/10/1911

APPENDIX

APPROVAL _____

RADIATION INCORPORATED

TEL: 1-314-651-4000 FAX: 1-314-651-4001

SOBAC, APT OF HARRISINTERTYPE CORPORATION, CLEVELAND, OHIO

FILE NAME DECODED

CONFIDENTIAL

113

100

SECRET (REF ID: A66544)

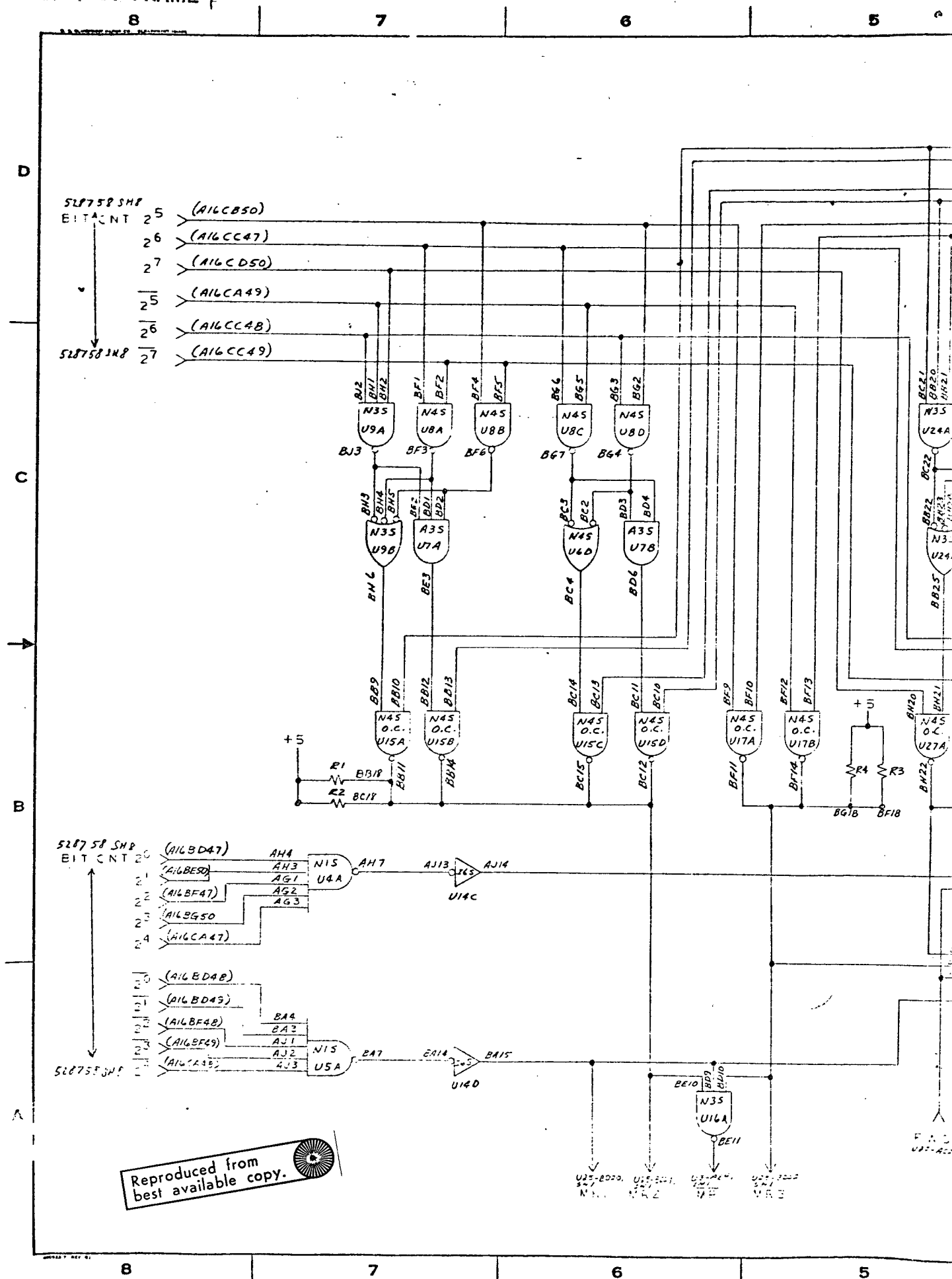
SIZE	CODE IDENT NO	RE
------	---------------	----

D	91417	528760	(
---	-------	--------	---

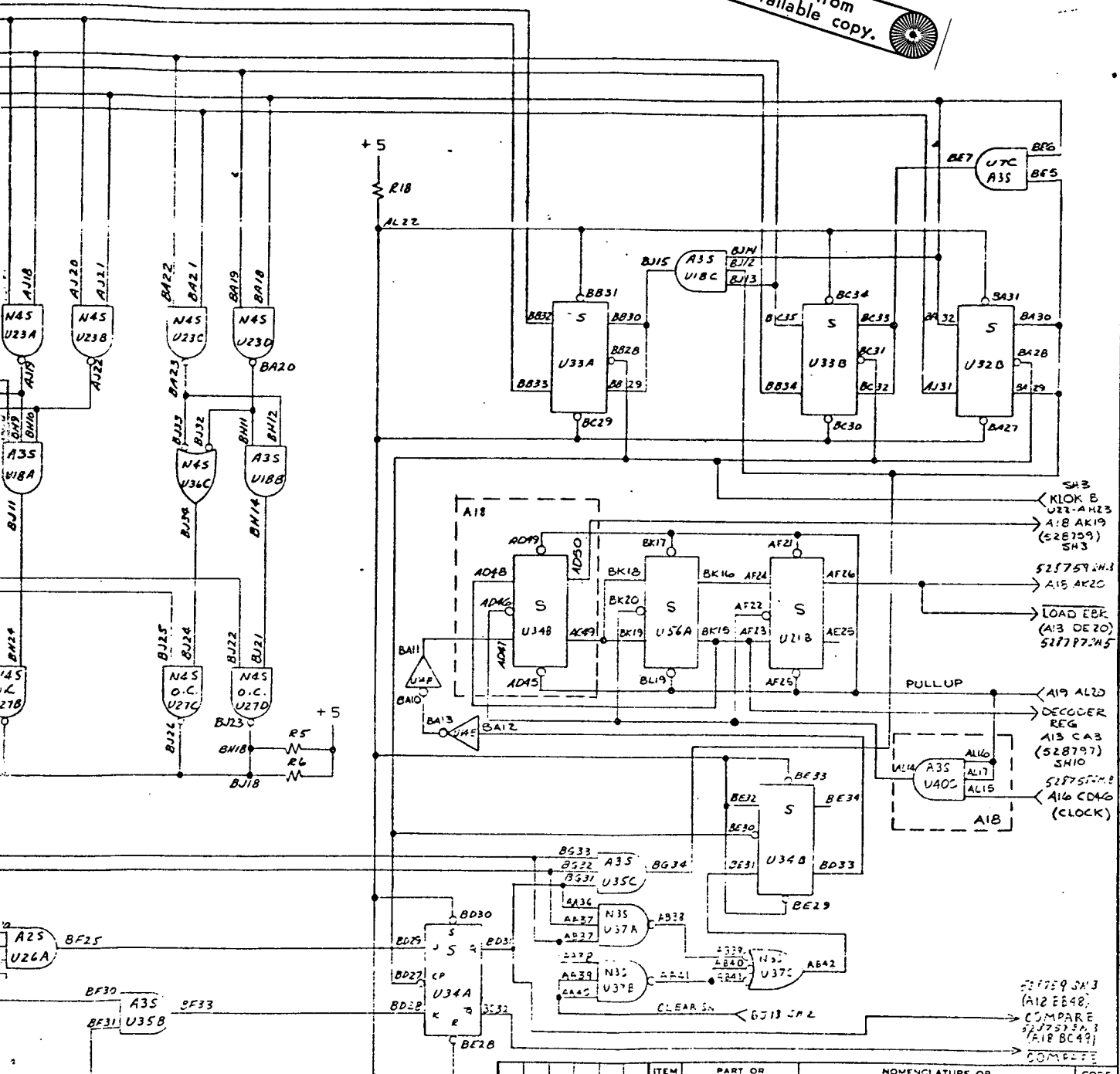
0	51417	
---	-------	--

A

FOLDOUT FRAME



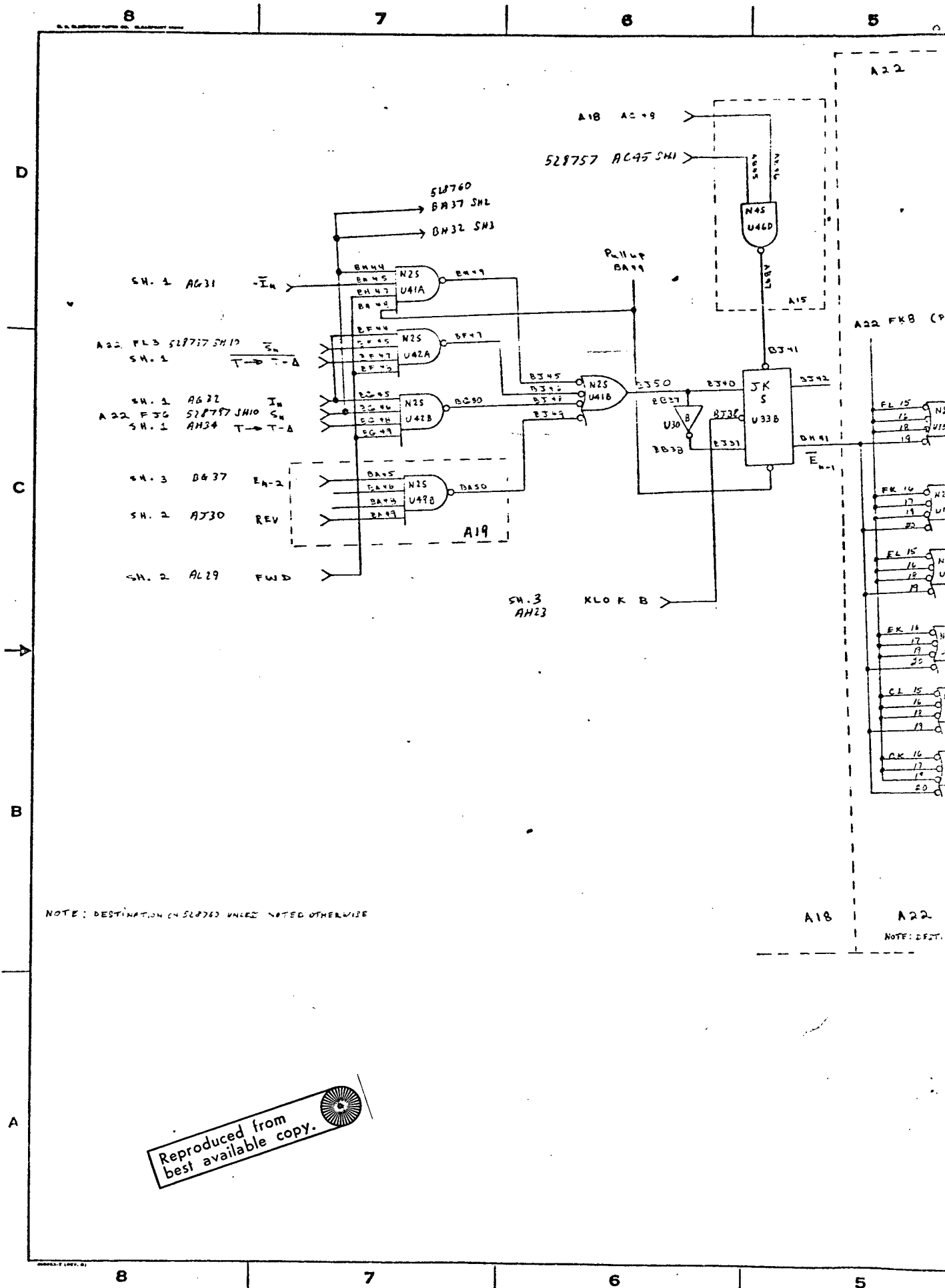
Reproduced from
best available copy.



REV CC
C47. AD50, SN1

BASH NO	NEXT ASSY	USED ON
APPLICATION		

				ITEM NO	PART OR IDENTIFYING NO	NOMENCLATURE OR DESCRIPTION	CODE IDENT
QUANTITY REQUIRED				LIST OF MATERIALS OR PARTS LIST			
UNLESS SPECIFIED ALL DIMENSIONS ARE IN INCHES APPLIED FINISH DIMENSIONS				RADIATION INCORPORATED SUBSIDIARY OF HARRIS-INTERTYPE CORPORATION, MELBOURNE, FLORIDA			
CONTRACT NO.				TITLE			
DR BY		CHK BY		FANUC SYSTEMS CONTROL SYSTEMS			
ENGINEER							
PROJECT ENGINEER							
APPROVAL				SIZE CODE IDENT NO		REV	
APPROVAL				D 91417		288760	
SCALE						SHEET 5	



1

3. $(-2\Delta + 1)$ indicates $\bar{S}_N I_N \cdot (\overline{T \rightarrow T - \Delta}) = 1$; $(-2\Delta + 1)$ is ANDed with Fwd in the Δ counter; should a forward move occur under this condition, the Metric will be debited by $-2\Delta + 1$ for passing through a double error, etc.

The control signals here then are anticipatory; they become effective if the forward or reverse move they anticipate occurs.

The main factor affecting the design here is the 20 MHz operating frequency. Using the TI 74S series logic, qualification of the counter stages is limited to five and sometimes six orders of logic; it is this limitation that suggests that the counter design give preferential treatment to its Fwd and Rev control lines. Generation of the Fwd and Rev signals by the control circuitry requires two orders of logic. So, propagation of these signals through gates in the counter should be kept to three orders or less. Now the generation of carries and borrows propagated to the Δ counter from the Mod- Δ counter depend on Fwd and Rev. Using Fwd and reverse in generating the carries and borrows would require propagating these signals through two more orders of logic; the carries and borrows then would be limited to one further order in the Δ counter. It is this restriction which prompted the generation of conditional signals such as F carry, R carry, etc. These conditional signals contain all the required terms except Fwd or Rev. They are then ANDed in the Δ counter as required.

A final consideration in the design of the Metric is the entry of the threshold changes.

When a threshold lowering occurs, the Metric will have value zero and Fwd will be true. What value the Metric assumes on the forward move depends on the syndrome bit.

When $S_N = 0$, the Metric must increment to $1 + \Delta$. Incrementing by +1 will occur automatically in the Mod- Δ counter. A gate qualified by $(T \rightarrow T - \Delta) \cdot \bar{S}_N$ enables the setting of 1Δ into the Δ counter.

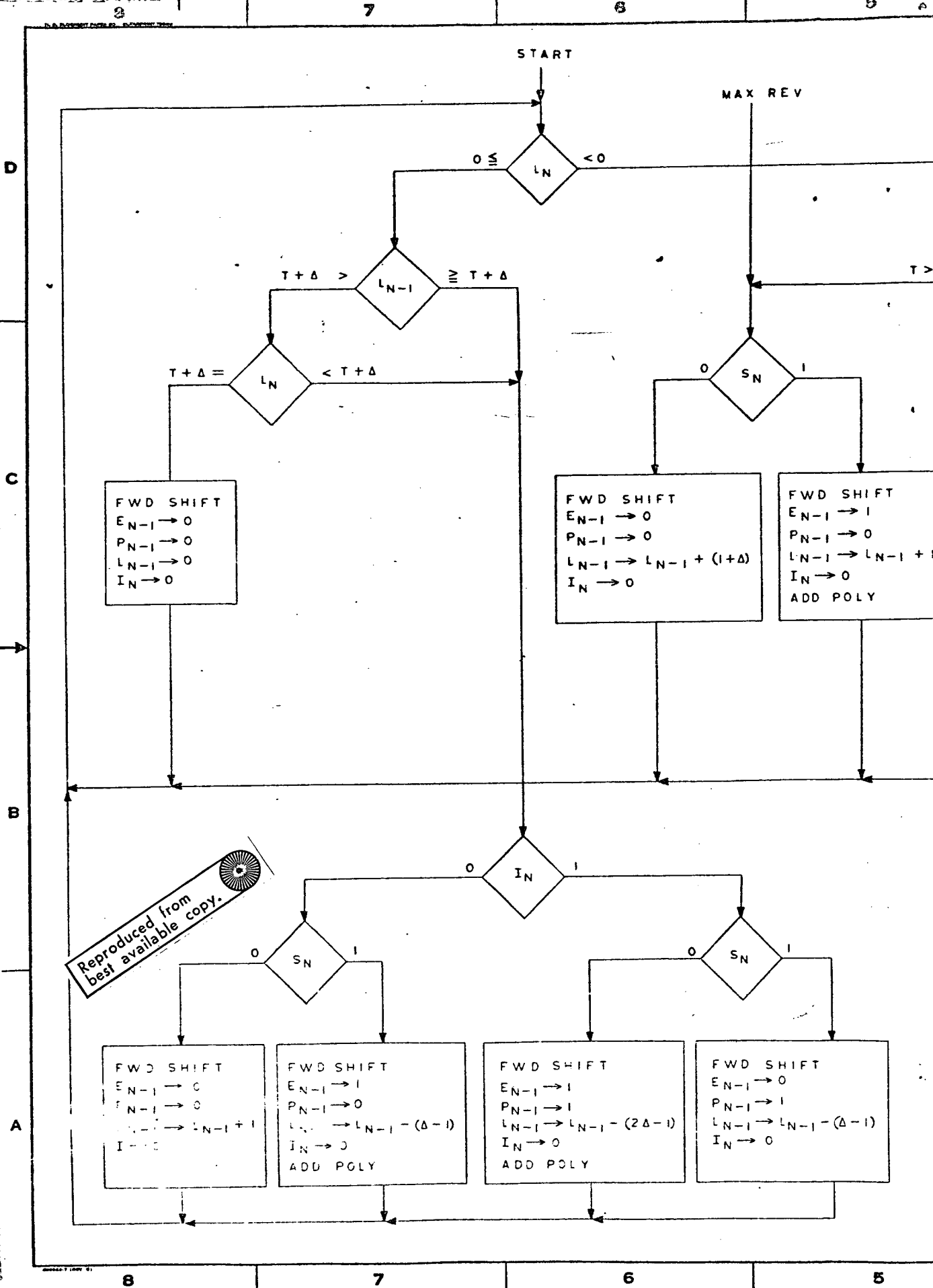
When $S_N = 1$, the Metric must increment to 1. This occurs automatically as above.

The Metric is implemented on drawing 528791, sheets 1 and 2.

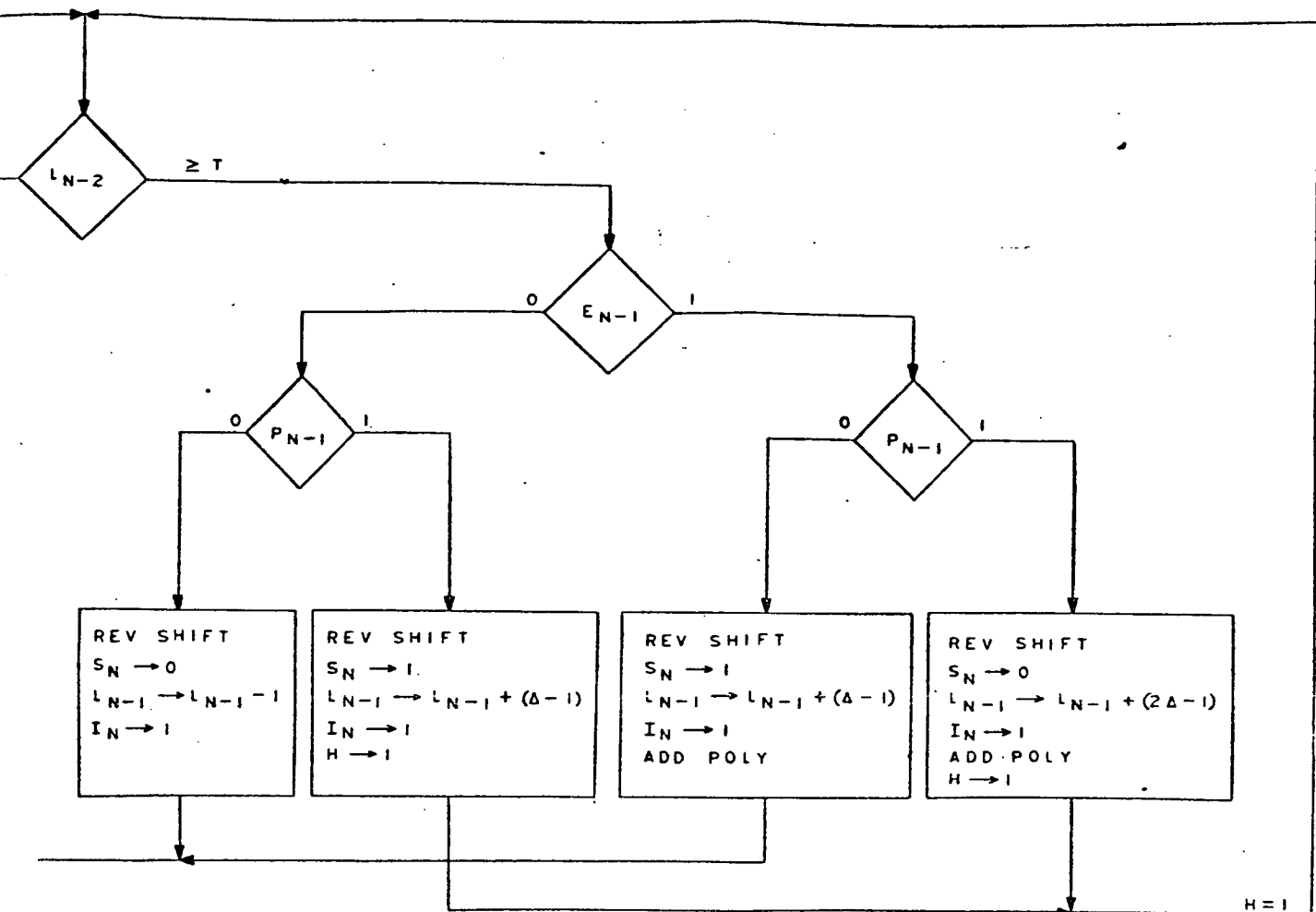
A flow chart diagramming decoder activity is given in Figure A2-3.4.1.2-3.

The term Max. Rev. introduced earlier had been explained only briefly. The details of its generation in the test set are given below. These details are given in context with a broader description of the decoder's operation and refer to the block diagram of Figure A2-3.4.1.3-4. So far, discussion has centered on decision making in generating the E_N , P_N pairs. Now the attention shifts to handling the pairs generated.

As forward progress is made through the tree, E_N , P_N pairs fill their respective registers. The contents of the two registers are then transferred to 8×32 -bit scratch pad memories.

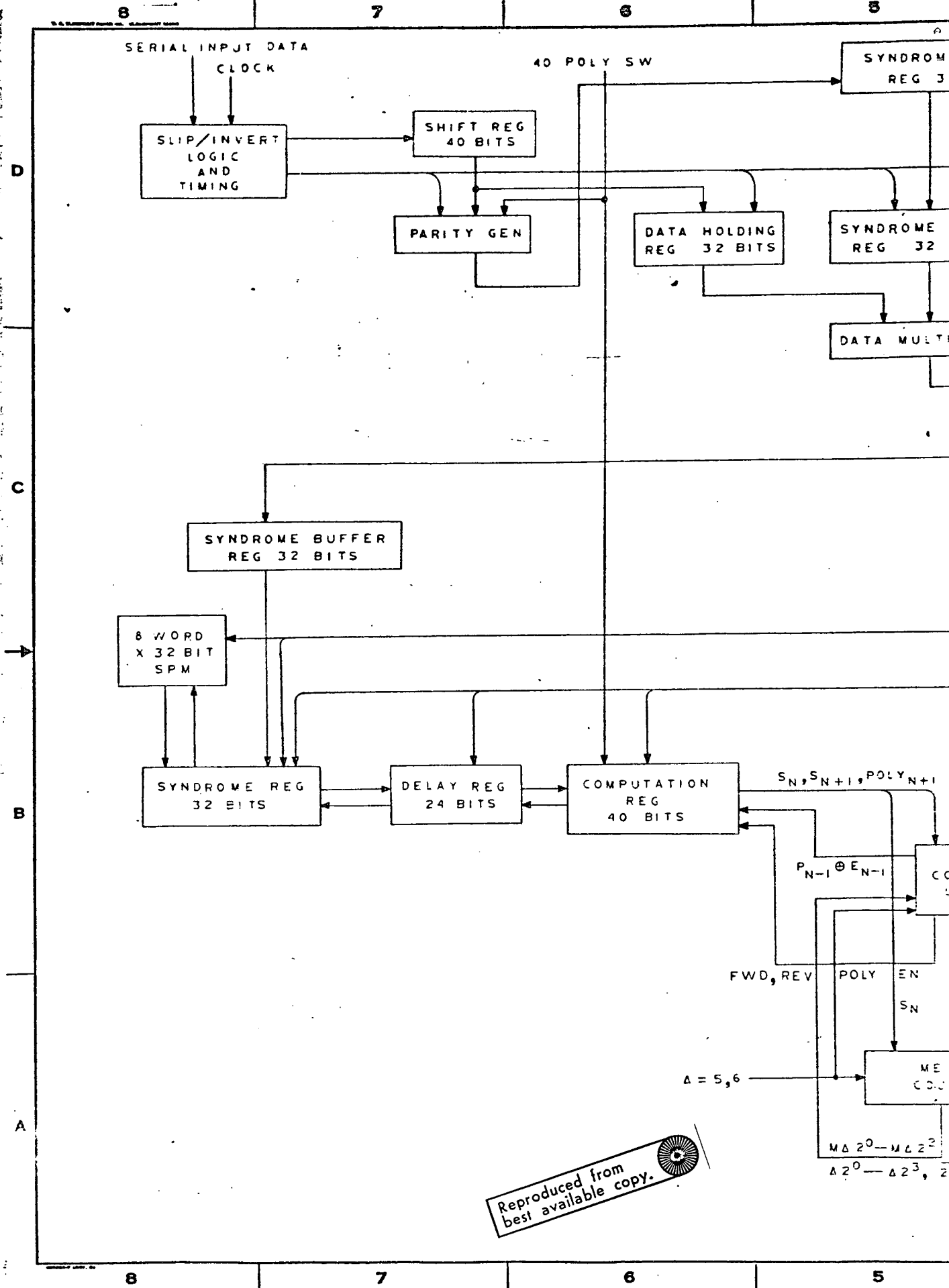


REVISIONS			
ITEM NO	DESCRIPTION	DATE	APPROVED

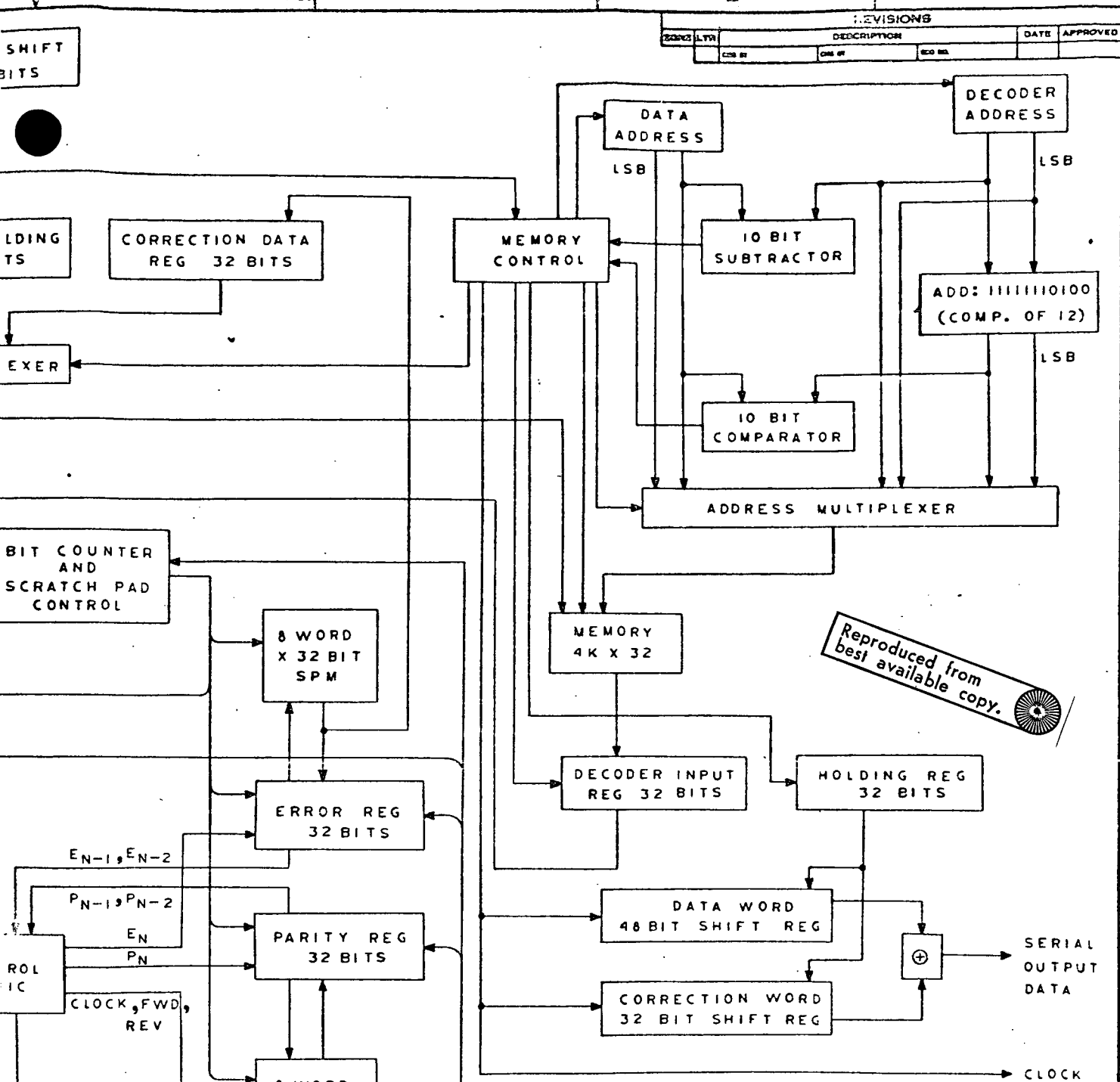


Reproduced from
best available copy.

QUANTITY REQUIRED		ITEM NO	PART OR IDENTIFYING NO	NOMENCLATURE OR DESCRIPTION	CODE IDENT
LIST OF MATERIALS OR PARTS LIST					
RADIATION INCORPORATED SUBSIDIARY OF HARRIS-INTERTYPE CORPORATION, MELBOURNE, FLORIDA					
PART TO BE FREE OF BURRS AND SHARP EDGES COMMERCIAL TOLERANCES APPLY TO STOCK SIZES MANUFACTURING TOLERANCES PER 900000 WORKMANSHIP PER 900000 SCREW THREADS PER MIL STD-9 ABBREVIATIONS PER MIL STD-12 LOGIC SYMBOLS PER MIL STD-806 DIMENSIONING AND TOLERANCING PER USAS T14.5-66 ELECTRICAL AND ELECTRONIC DIAGRAM PER USAS T14.15-66 GRAPHIC SYMBOLS FOR ELECTRICAL AND ELECTRONIC DIAGRAM PER USAS T132.2-67 ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER USAS T132.10-65 RELAYING SYMBOLS PER MILS A2.0-58			CONTRACT NO DR BY CHE BY ENGINEER MAJNAR PROJECT ENGINEER APPROVAL APPROVAL		
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES AND INCLUDE APPLIED FINISH TOLERANCES 2 PLACE DIM. 3 PLACE DIM. ANGLES			TITLE FAND DECODER FLOW CHART		
APPLICATION PART NO NEXT ASSY USED ON			SIZE D		
4			CODE IDENT NO 91417		
3			Figure A2-3.4.1.2-3		
2			SCALE		
1			SHEET		



Reproduced from
best available copy.



Reproduced from
best available copy.

QUANTITY REQUIRED		ITEM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	CODE IDENT
LIST OF MATERIALS OR PARTS LIST					
CONTRACT NO.			RADIATION INCORPORATED		
SUBSIDIARY OF HARRIS-INTERTYPE CORPORATION, MELBOURNE, FLORIDA			TITLE		
ENGINEER: MALNAR			FANO DECODER		
PROJECT ENGINEER			BLOCK DIAGRAM		
APPROVAL			SIZE: D	CODE IDENT NO: 91417	REV: Figure A2-3.4.12-4
APPROVAL			SCALE	SHEET	

PART TO BE FREE OF BURRS AND SHARP EDGES		UNLESS OTHERWISE SPECIFIED	
CONFERENTIAL TOLERANCES APPLY TO STOCK SIZES		DIMENSIONS ARE IN INCHES	
MANUFACTURING TOLERANCES PER 900000		AND INCLUDE APPLIED FINISH	
WORKMANSHIP PER 900000		TOLERANCES	
SCREW THREADS PER MIL-STD-8		2 PLACE DIM.	3 PLACE DIM.
ABBREVIATIONS PER MIL-STD-12		±	±
LOGIC SYMBOLS PER MIL-STD-400		ANGLES	
DIMENSIONING AND TOLERANCING PER USAS		±	
Y14.5-66		±	
ELECTRICAL AND ELECTRONIC DIAGRAM PER USAS		±	
Y14.15-66		±	
GRAPHIC SYMBOLS FOR ELECTRICAL AND		±	
ELECTRONIC DIAGRAM PER USAS Y32.47		±	
ELECTRICAL AND ELECTRONIC REFERENCE		±	
DESIGNATIONS PER USAS Y32.15-65		±	
WELDING SYMBOLS PER AWS A2.6-58		±	

NEXT ASSY	USED ON
APPLICATION	

In the reverse mode, E_N , P_N pairs are converted into syndrome bits. If this mode prevails, the reformed syndromes move back through the computation register and delay register and fill the syndrome register. There they are parallel transferred into the 8 x 32-bit syndrome(s) scratch pad memory.

In the searching process, Max. Rev. will be set to a one when the decoder, in the reverse mode, withdraws all bits available from the error and parity scratch pads. At this time, the decoder will have reached branch $N = 1$ and threshold lowering will occur as per the earlier flow chart of Figure A2-3.4.1.

The process of adding bits to or removing bits from the scratch pads is dictated by the scratch pad control logic. This circuitry consists mainly of two counters, two comparators and some control flip-flops.

One counter called the bit counter uses Fwd., Rev. and the decoder clock to detect when the parity and error shift registers have filled with 32 bits and are ready for a parallel transfer to the scratch pads or when the registers, in the reverse mode, have empties and require a parallel transfer from the scratch pads.

The bit counter has 8 stages; the first 5 count the bits shifted into the E and P registers. The last 3 address the scratch pads. In the forward mode, one of the control flip-flops is qualified by count XXX01111. This flip-flop then enables a transfer to the E and P scratch pads on count XXX10000. In the reverse mode, one of the control flip-flops is qualified by count XXX10001; this flip-flop then enables a transfer to the E and P registers on count XXX10000.

Use of the bit counter to effect loading the syndrome register is similar to the process above. As mentioned earlier, a third scratch pad called the syndrome scratch pad provides storage space for bits during the reverse mode. Once the decoder switches to the forward mode, the bits stored in the S scratch pad are transferred back to the S register. When the scratch pad empties, new syndromes from the plated wire buffer memory are loaded into the S register via the S buffer register. To detect whether the S register should be loaded from the scratch pad or from the buffer register, a reference counter and a comparator are used with the bit counter. To see how this works, first note that in an error free situation, the E and P scratch pads are full and the three most significant bits of the bit counter compare with the 3 bits of the reference counter. Note also that, as the decoder continues forward, this comparison remains valid. Now, when an error situation causes the decoder to reverse itself and causes a transfer from the scratch pads to the E and P registers, the scratch pad address will decrement and no longer compare with the reference counter. This compare or no-compare condition is then the factor determining whether the S register is loaded from the S buffer register or the S scratch pad, respectively. In the logic, a control flip-flop is qualified by bit count XXX01111 ANDed with Fwd. and compared so that, in the forward mode, this flip-flop enables a transfer from the S buffer register to the S register on bit count XXX10000. A different control flip-flop is qualified by bit count XXX01111 dotted with a Fwd and compare to effect loading the S register from the S scratch pad on count XXX10000.

Loading of the S scratch pad is coincident with the transfer of data to the E and P registers. The control flip-flop dedicated to loading the S scratch pad is qualified by XXX10001 ANDed with Rev.

From the simplified block diagram of Figure A2-3.4.1.2-5, one can note the relationships of the scratch pad control logic and the manner in which Max. Rev. is detected. A second comparator compares the address bits with the reference counter bits and detects when the E and P scratch pads are about to empty. The comparator will indicate Max. Rev. when the address count is one greater than the reference count. At this point, a word about the comparators might be helpful. The comparators compare the 2's complement of one number with the 1's complement of the other. For instance, in generating the compare signal used two paragraphs above, a comparator will qualify the compare flip-flop when the scratch pad address is one less than the reference count. When this relationship exists, the 1's complement of the reference count will equal the 2's complement of the scratch pad address.

The logic of the two comparators is given as sheet 4 of the decoder control logic. The logic for the control flip-flops and drivers is given on sheet 3, drawing 528759. The bit counter logic is given on sheet 8 of drawing 528758.

Bit Counter

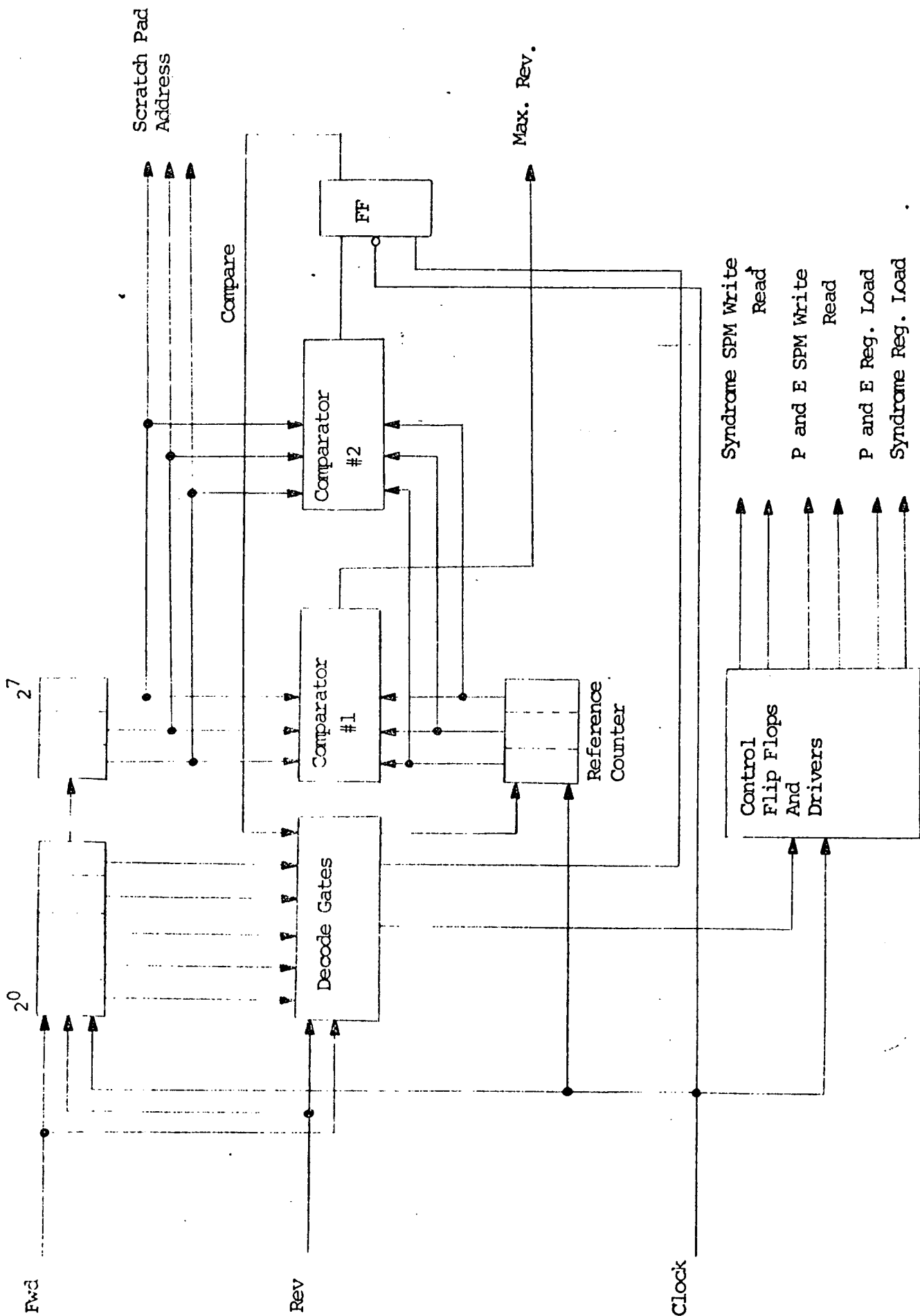


Figure A2-3.4.1.2-5. Scratch Pad Control

A2-3.5 Group Synchronizer and Demultiplexer - A8A1A2

The group synchronizer accepts data and clock (information rate) from the decoders and then identifies prime frames, words and bits. The incoming bit stream is converted from serial to parallel and then presented to the correlator and demultiplexer. During the process of identifying specific input patterns, the group sync also generates control signals for use elsewhere in the system. In addition, the unit continuously selects data samples from the input data stream and, as long as the data patterns are recognized, maintains the system in a sync lock condition.

A2-3.5.1 Group Sync Policy

The overall design specifications for the group synchronizer are listed below:

1. Bit Rate - 9 MHz max
2. Frame Length - Fixed from Transmitter Rate
3. Frame Sync Code - 30 Bits (Fixed Length)
4. Frame Sync Strategy - 2 Modes
 - a. Search - Fixed, 1 compare to Lock 0-4 Errors Selectable
 - b. Lock - Fixed, 1 no compare to Search 0-10 Errors Selectable
5. No Aperture Width - 1 Bit
6. Output Signals
 - a. Frame Lock Level
 - b. Parallel Data to Demux

A2-3.5.2 Group Synchronizer Block Diagram

The basic group synchronizer consists of a 30-bit shift register, a parallel digital correlator, and timing and control logic. However, since the functional relationship between any Group Sync and Demux is basic to their operation many of the functions serve both the group sync and demultiplexer. Therefore, for the purpose of this discussion they will be considered as an integral unit (Figure A2-3.5.2).

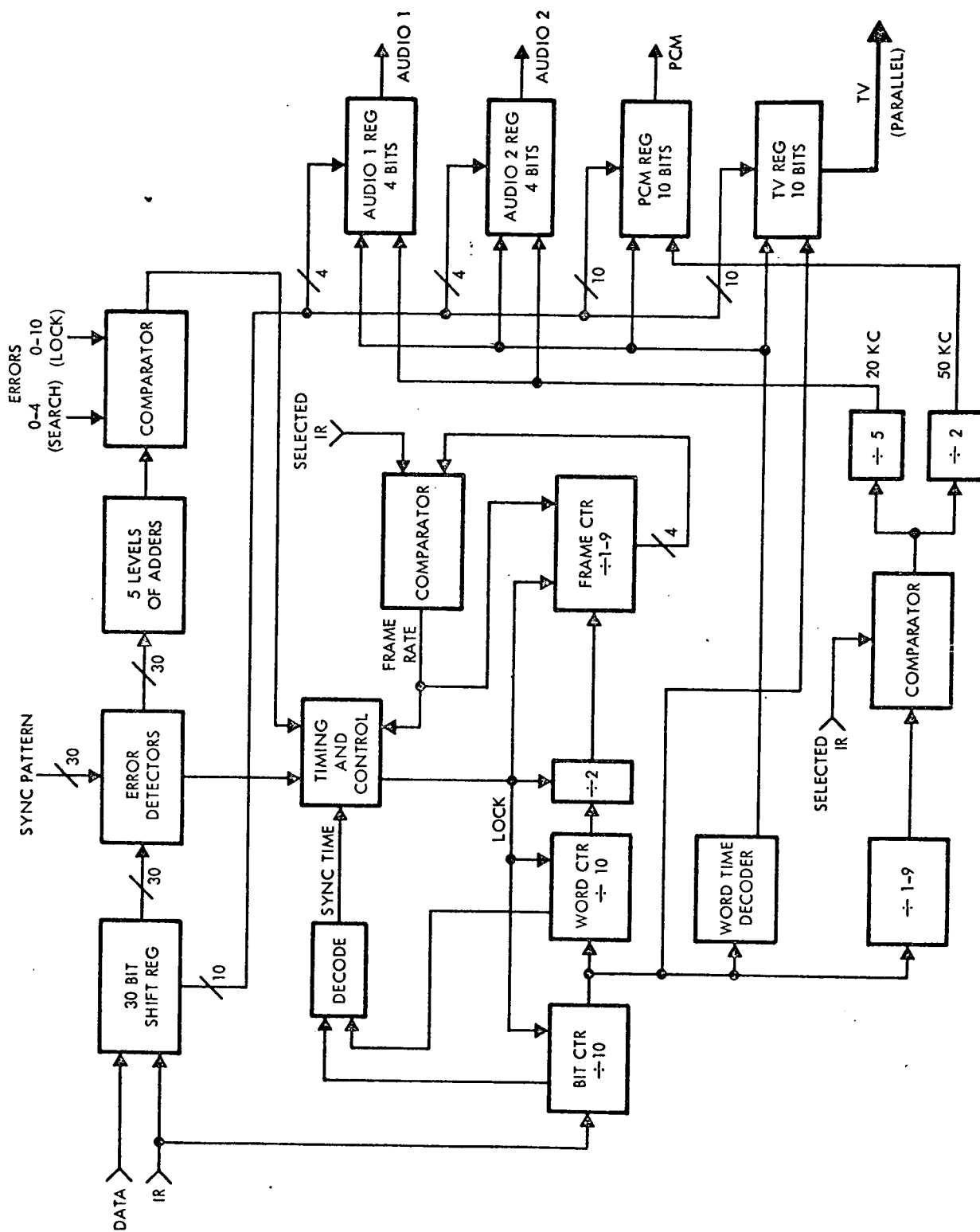


Figure A2-3.5.2. Block Diagram - Group Sync and Demux

The shift register consists of a 30-bit shift register wired for serial in/parallel out operation. The register accepts serial data from the decision device and stores the last 30 bits received. The output of each stage is presented as an input to the Parallel Digital Correlator.

Within the Parallel Digital Correlator, two processes take place. First a binary number equal to the number of differences between the expected sync pattern and the contents of the 30-bit register is obtained. This is accomplished by comparing the expected sync pattern to the data in the shift register by using an exclusive "OR" circuit. Any differences in the patterns will be added and stored through five levels of addition. The fifth level of adders contains the binary number of the differences in patterns. Thus, the second process takes place, the difference between this binary number and the number of allowed errors is obtained. If the value of the difference is less than or equal to the number of allowable errors, then correlation is achieved.

Once correlation is achieved, the Group Synchronizer signals timing and control that it is in sync. Previous to correlation the bit and word counters had been set to a specific state. On correlation they are allowed to start counting. As long as the group sync is in lock these counters in conjunction with timing and control will now merely output the last ten stages of the 30-bit shift register to the appropriate destination. If for any reason the group sync loses "lock" the processing of data is immediately terminated.

In addition to keeping track of bits and words, the Group Sync and Demux also count each frame. The frame counter is used to maintain timing and also to ensure that synchronization is maintained. It does this by signaling timing and control that a frame has elapsed. This ensures that the group sync checks its synchronization once each frame. If the number of errors in the pattern is less than or equal to the number of allowable errors (0-10) then the system is maintained in "lock"; otherwise the processing of data is terminated, and sync search is reinstated.

A2-3.6 Receiver Rate Buffer (A8A1A1, A8A1A8, and A8A1A11)

The receiver rate buffer accepts 20-bit TV data words from the demultiplexer, detects vertical TV sync words, formats the TV synchronization and buffer synchronization words, and writes assembled 30-bit words into the plated wire memory. Thirty-bit words are read out of memory as a function of TV video reconstructor operation. The 30-bit words are disassembled into 10-bit words and sent to the video reconstructor.

A2-3.6.1 Receiver Buffer Operation

The receiver rate buffer consists of a random access, 30 bit x 512 word, 200 nano-second cycle time plated wire memory with associated input - output controls, buffer synchronization logic, and input data formatting logic as shown in Figures A2-3.6.1-1.

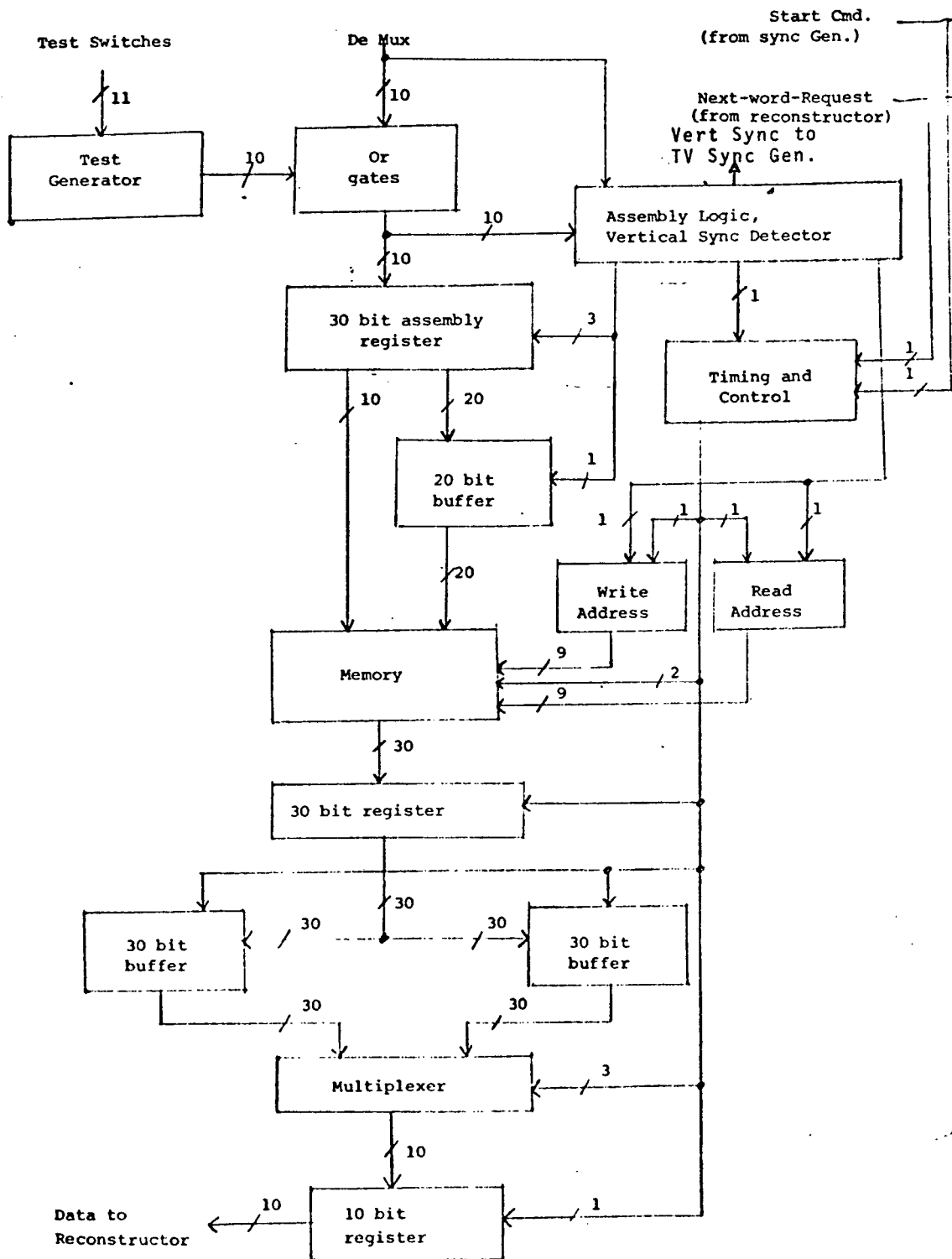


Figure A2-3.6.1-1. Rate Buffer

Referring to Figure A2-3.6.1-1, data from the time division demultiplexer passes through a set of gates and is assembled into 30-bit words. Assembly aligns the data so that TV sync and buffer sync words occupy the same bit positions at the receiver as they did at the transmitter.

Once 30 bits of data are assembled, a write request is made to the timing and control logic by the assembly logic and simultaneously bits 1-20 of the 30 bits are moved from the 30-bit assembly register into the 20-bit buffer. While the request awaits service by the Timing and Control (T&C), new words can be loaded into bits 1-20 of the assembly register. By the time these bit positions are filled T&C will have commanded the memory to accept the data at its inputs. Data is removed from the memory on an as-needed basis, 10-bit words are transferred from the 10-bit register to the Reconstructor or to the Statistical Decoder depending on the mode of operation. The operating unit issues a next-word-request, this pulse is gated through the T&C to the 10-bit register where it loads in a new word from the multiplexer. In the T&C the same pulse increments a mod-6 counter so that the multiplexer will select the next word to be accepted by 10-bit register.

The mod-6 counter is decoded to tell the T&C when one of the two 30-bit buffers is empty. When such a situation is indicated the contents of the 30-bit register is transferred into the empty 30-bit buffer. Under worst-case conditions, one of the 30-bit buffers will be emptied and refilled by the contents of the 30-bit register, then before the 30-bit register can receive new data from the memory, the second 30-bit buffer's data and the first 20-bits of the first buffer's data may pass through the multiplexer. This case occurs when:

1. The ZOI/ZOP mode is selected and
2. Each of the time tags represents one sample count (at 9 MHz) and
3. The read request decoded from the mod-6 counter occurs at the beginning of a memory write cycle.

The basic operation of the receiver buffer can be summarized as follows:

1. Write commands and read commands are decided on a priority basis; if a write request and read request are received simultaneously, the T&C will generate a read command.
2. The input section (consisting of the input OR gates, test generator, assembly logic, assembly register and 20-bit buffer) formats data at the memory input and creates write requests. The write request generates a write command to the memory and the data is absorbed.
3. The output section (consisting of the 30-bit register, two 30-bit buffers, the multiplexer and 10-bit register) unpacks the 30-bit words for the requesting unit and smooths the data flow between the memory and the requesting unit.

4. The control section consisting of the T&C, and write and read address counters service the input and output sections. When write commands are issued the T&C holds the memory mode line low and generates a cycle initiate command (MRQ). At the end of the write cycle the T&C updates the write address. When read commands are issued, the T&C holds the memory mode line high and generates a cycle initiate command (MRQ). At the end of the read cycle the T&C updates the read address.

A test generator is provided in the receiver buffer as a checkout aid. The generator can:

1. Generate a black and white bar pattern (each of the eight bars has a 6-bit time tag of 59) and
2. Generate a horizontal sync word at the end of the bar pattern and
3. Generate a buffer sync word following the horizontal sync word (each of the buffer sync bits are controllable from test switches on panel A1A14) and
4. Generate one or two lines of TV data per line read out of memory (selection here is controllable by the test sequence repetition switch on panel A1A14).

Note that the test generator was designed for use in the ZOP/ZOI mode only.

A2-3.6.2 Buffer Synchronization

The purpose of the Buffer synchronization operation is to assure the proper relationship between the state of fill of the transmitter and receiver buffers. This can be accomplished by controlling the overall delay through both buffers, since errors in buffer fullness relations appear as time errors in the output data.

Buffer synchronization is established with the aid of the assumption that the transmitter buffer is empty at the end of vertical blanking, a reasonable assumption since at that time the buffer has just experienced a long period of unload operations, with no load operations.

At the end of vertical blanking, a vertical sync word is inserted at the transmitter. Recognition of this word at the receiver causes the following events:

- a. The TV Sync Generator is preset to the beginning of vertical blanking.

- b. The buffer read and write addresses are both set to zero (i.e., the buffer is set to empty).
- c. Buffer unload and TV reconstruction operations are suspended until the end of a fixed time delay. For convenience, this delay was selected as the vertical blanking interval, and is generated by the TV Sync Generator.

The effect of these operations is to generate a fixed overall time delay. At a time when the delay in the transmitter buffer is assumed to be zero, a delay in the receiver buffer equal to the vertical blanking interval is generated by starting from an empty buffer and filling it while inhibiting outputs during this time interval. At other times during the frame this delay will be distributed between the transmitter and receiver buffers depending upon data activity, but the overall delay will be the same, barring time tag errors.

The effective buffer size with this technique is equal to the amount of data transmitted during the vertical blanking interval, and therefore varies with the transmitted information rate. At the highest information rate this amount of information does not exceed the size of the installed memory.

Note that this procedure also performs the TV synchronization operation.

A2-3.7 TV Data Reconstructor - A8A1A7

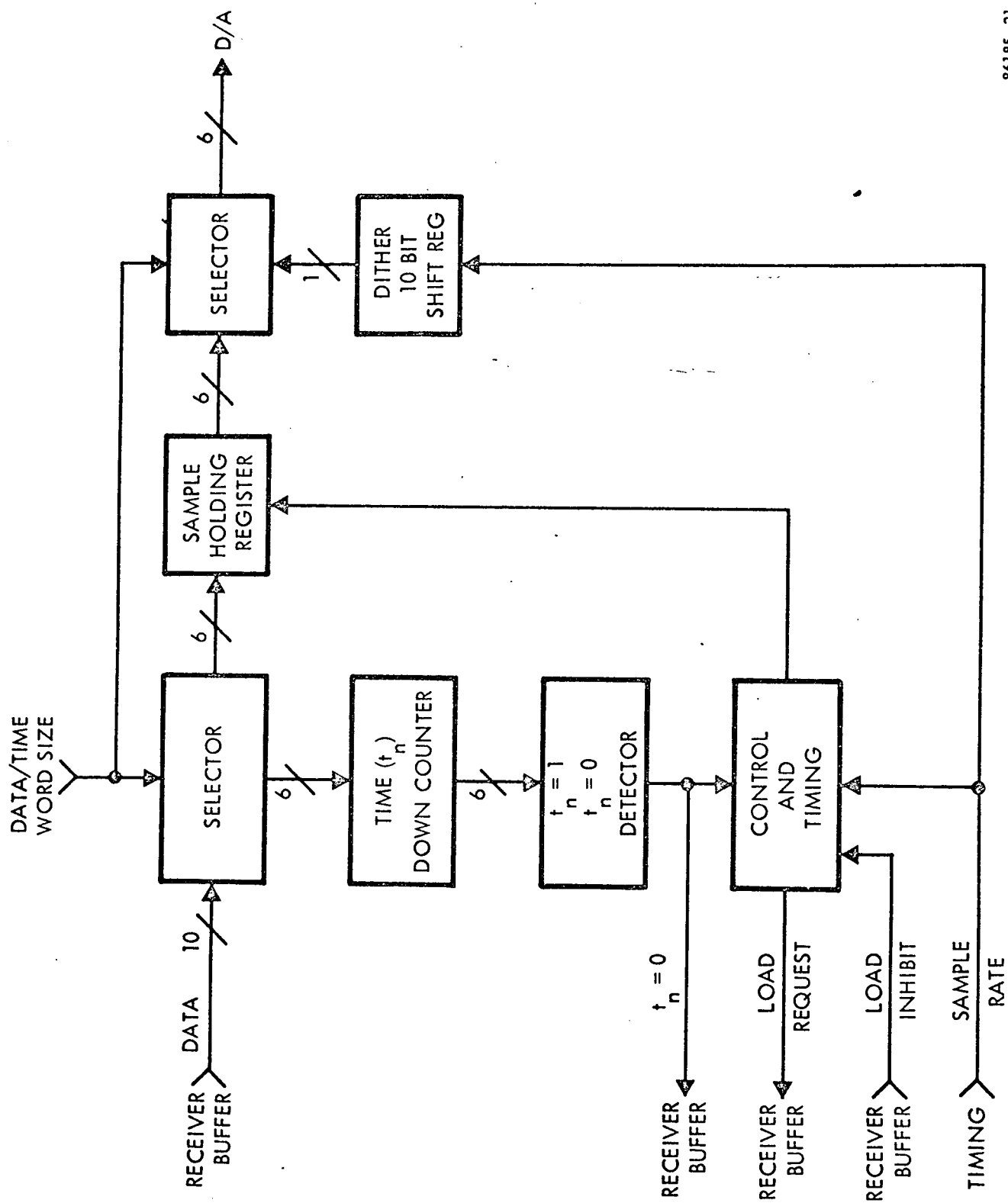
The TV data reconstructor obtains reduced TV data from the receiver rate buffer. This information is in the form of a sample and the number of times the sample should be repeated (zero order reconstruction) for both ZOI and ZOP reduction techniques. Thus, reconstructor operation is independent of reducer mode. The received sample is presented to a D/A converter for the required number of sample times producing a reconstructed analog video waveform. Amplitude dither is combined with the received data sample prior to the D/A input to reduce false contouring in the reconstructed TV picture.

Reconstruction operation is inhibited during TV blanking when TV sync is being added to the output waveform. Normal reconstructor processing is also inhibited when the receiver buffer becomes empty.

Reconstructor Operation

A block diagram of the TV data reconstructor is shown in Figure A2-3.7.

Reconstructor operation basically consists of outputting received nonredundant samples to a D/A converter. The received sample is obtained from the 10-bit sample register



86185-21

Figure A2-3.7. TV Data Reconstruction Block Diagram

in the receiver rate buffer section. Gating controlled by a three position switch on the receiver front panel determines how many of the 10 bits are sample information and how many are time data. With the sample size switch in the 6-bit position, 6 of the 10 bits are loaded into the 6-bit reconstructor sample register; the 5-bit position loads 5 bits; the 4-bit position loads 4 bits. The remaining bits in each case are loaded into the 6-bit reconstructor time (t_n) down counter. After sample and time loading is complete, the t_n counter is decremented at the TV sample rate which is selected on the front panel. During t_n down counting the sample register 6-bit output is presented to a D/A converter for conversion to a voltage level. When the t_n down count reaches 1, the load enable flip-flop is qualified by detecting the 1 state with a NAND gate. This detection is performed on the 5th MSB of the counter so that both 1 and 0 are detected allowing the flip-flop to remain qualified for the counts of 1 and 0. The next sample clock that decrements the t_n counter to zero also strobes the load enable flip-flop causing its output to change state. This allows the sample rate clock to load the sample register, load the down counter, and begin the process in the receiver buffer that loads the next 10-bit word into the 10-bit register. If a time tag other than 1 or 0 is loaded into the t_n down counter, the 1 and 0 detect gate turns off, allowing the load enable flip-flop to be cleared. No more loads are allowed until a count of 1 has been reached. If the time tag that is loaded into the t_n counter is 1, the load enable flip-flop remains qualified and the next clock pulse that decrements the counter to 0 begins another load cycle. In all cases loading is completed before the next clock pulse occurs.

Error-free transmission of data means zero time tags never occur in the received video data. Only TV sync words and blanking fill words have zero time tags. When transmission noise does cause errors in the received data, it is possible to obtain zero time tags in received words that will not be recognized as "special" words. The reconstructor protects against the reception of zero time data by detecting the zero state with a NAND gate and inhibiting any further down counting until the zero state is removed by the next t_n counter load. This causes no change in normal operation since decrementing the counter to 0 begins a load process. If a zero time tag is loaded, the 1 and 0 detect gate thinks it sees a 1 and the load enable flip-flop remains qualified. The next clock pulse is not allowed to decrement the counter causing the load flip-flop to think the counter has been decremented to zero and allowing the sample rate clock to begin the next load process. In effect, the zero detect gate forces the counter to treat a zero time tag like a 1 time tag.

The sample rate clock also strobes the 10-bit dither feedback shift register. Each bit of the 1023 bit pseudo random sequence generated by this shift register is modulo 2 added to a selected bit of the sample register. This addition takes place at the output of the 6-bit sample register; therefore, the D/A converter receives a TV data sample plus amplitude dither. The bit position for the modulo 2 addition is chosen by the sample size switch on the front control panel of the receiver. Dither is not added to any bit if 6-bit samples are being processed. Five-bit sample operation allows dither to be added to the LSB of the 6-bit register output. Dither is added to the 5th MSB if 4-bit sample processing is taking place. The 1023-bit pseudo random sequence is obtained from the 10-bit shift register by modulo 2 adding the output bit (10th bit) and the 7th bit of the shift register and feeding this result to the shift register input. The zero state is

detected in the shift register, and this result is "OR'ed" with the feedback into bit number 1. The zero detector provides protection against becoming locked in the zero state.

No special starting procedures are required for the TV data reconstructor. When operation begins, the dither shift can begin in any state. The t_n counter will decrement to 1 and begin loading procedures independent of its beginning state. Also the load enable flip-flop will be set to the correct state by the first sample rate clock pulse. Some initial incorrect data may be sent to the D/A converter, but the system will correct itself the first time a count of zero is reached in the t_n counter.

Various external controls on reconstructor operation exist. When the receiver buffer becomes empty, further unloading of the buffer is inhibited. The reconstructor continues to operate on the nonredundant sample data in the two 30-bit buffers until buffer emptiness indication disappears. In effect, the reconstructor never knows that the buffer is empty, for its operation continues in a normal manner. Also, sample register and t_n counter are inhibited by the detection of TV sync in the 10-bit register in the receiver buffer subsystem. The Reconstructor is restated by signals from the TV Sync Generator which occur at the end of vertical and horizontal blanking.

A2-3.8 Statistical Decoder - 2A8A6A1 & 2

The Statistical Decoder accepts data from the Rate Buffer and Sample Rate pulses from the Timing Generator, detects code words in the data, processes in accordance with these code words and regenerated digital words representing the TV video. These output words are transmitted to the Digital to Analog Converter.

A block diagram of the Decoder is shown in Figure A2-3.8. Data from the Rate Buffer is loaded in parallel into the 10-bit P Register. The data is then serially shifted into the Q Register. The P Register is automatically reloaded after the last data bit is shifted from it to the Q Register. Meanwhile, a series of \emptyset pulses are generated by the \emptyset Generator, initiated by a Sample Rate pulse from the Timing Generator. The \emptyset pulses and the data from the Q Register are applied through the Code Patching to the Code Detect Logic. The Code Patching determines the interpretation made by the Decoder of particular bit pattern received, and corresponds to the code selection patching of the Statistical Encoder in the Transmitter. Detection of any code halts the shifting of the P and Q Registers, thus aligning the data in the Q Register for the word length being decoded. At the same time, the particular code detected is stored in the C' Register, and subsequently transferred to the C Register. The detected code may require updating the absolute value of the video, or adding a positive or negative delta to the previous value. Consider first the case of updating the absolute value. Data from the Q Register, already properly word aligned, is transferred to the A' Register. This data is LSB justified by the QL Logic so that deltas may be added in a fixed position independent of Quantization Level being processed. The current quantization level (4, 5 or 6 bits) is detected and stored in QL Logic. The

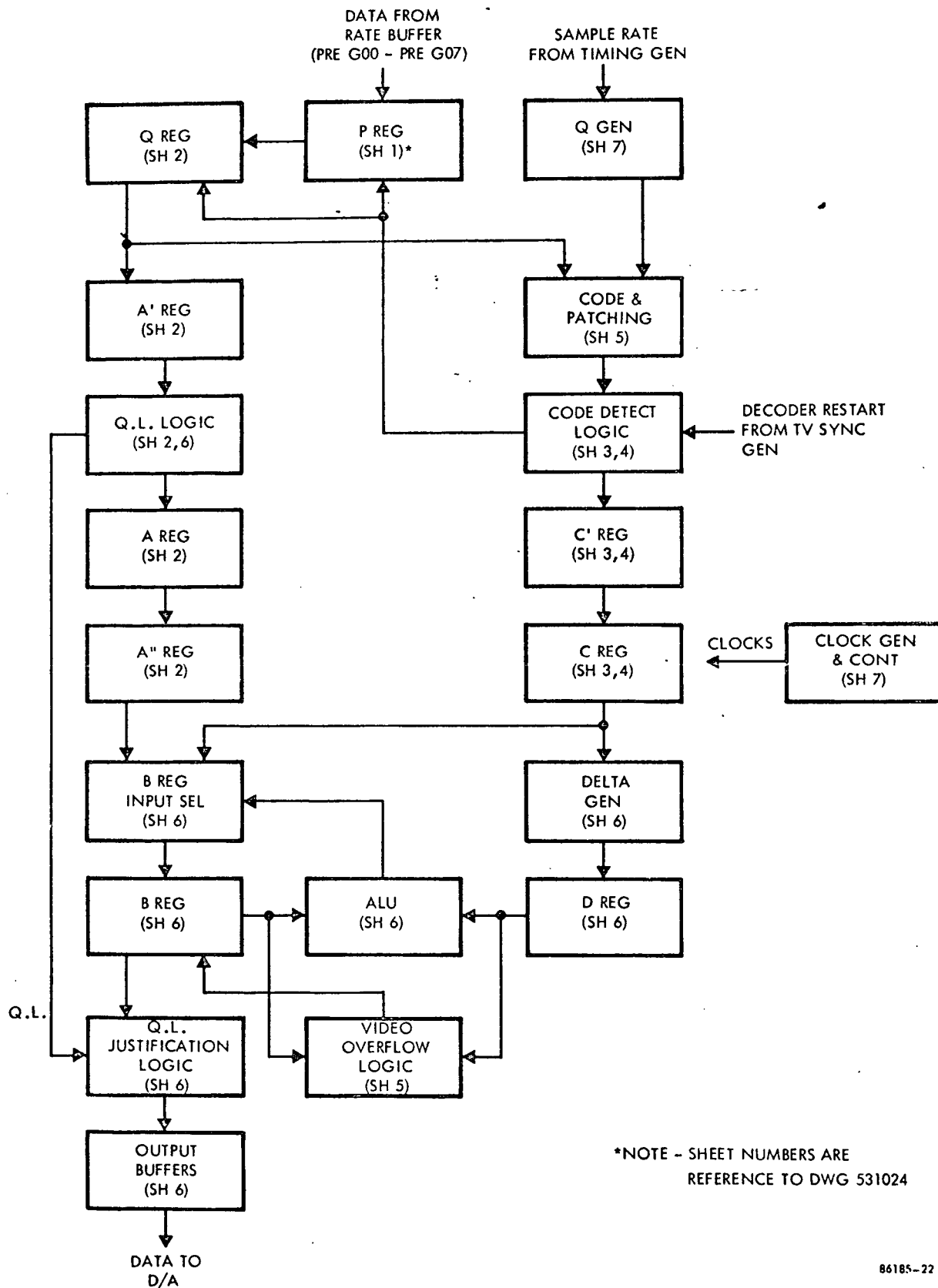


Figure A2-3.8. Statistical Decoder

LSB justified data is then stored in the A Register and subsequently transferred to the A'' Register. The output of this register forms one input to the B Register Input Selector, controlled by an output from the C Register. For the case under discussion, the detected code in the C Register calls for an update of the absolute value, so the data from the A'' Register is selected for transfer to the B Register. The data in the B Register has been LSB justified by the QL Logic, and must be MSB justified for the Digital to Analog Converter. This operation is performed by the QL Justification Logic, controlled by QL signals from the QL Logic. The data is then buffered and sent to the D/A Converter.

Consider now the case where a delta is to be added to the previous video values stored in the B Register. In this case, the control line from the C Register to the B Register Input Selector will be in the state selecting the ALU (Arithmetic Logic Unit), rather than the A'' Register as the B Register input. The code stored in the C Register is applied to the Delta Generator which produces the required delta in binary form, and stores it in the D Register. The contents of the B and D Registers are algebraically added to produce the new value for storage in the B Register.

The contents of the B and D Registers are monitored by the Video Overflow Logic, and delta additions which would cause the B Register to overflow or underflow are prevented by inhibiting updates of the B Register. Such overflow or underflow can occur as a result of link errors, and causes large brightness errors on the TV screen.

Detection of either a horizontal or vertical TV sync code by the Code Detector Logic indicates that the end of the current line (or frame) has been reached and initiates a special sequence of events:

- a. The next word, an unused 8-bit buffer word is discarded.
- b. Six bits of absolute value are transferred to the B Register.
- c. Two bits of Quantization Level value for the next line are decoded and stored in the QL Logic.
- d. Three data words are decoded, to fill the A', A, A'', C', C and D Registers with valid data.
- e. The Decoder halts until it receives the Restart signal from the TV Sync Generator.

Following the restart, the normal processing operations described above are carried out.

A2-3.9 TV Sync Generator - A8A1A10

The purpose of the TV Sync Generator is to generate TV blanking and synchronization signals coherent with the reconstructed TV video from the TV Reconstructor or the Statistical Decoder.

A block diagram of the Sync Generator is shown in Figure A2-3.9. Basic timing is derived from a one Megahertz reference signal derived from the incoming information rate by the Timing Generator. This signal is frequency multiplied to 9 MHz in a phase-lock multiplier to provide a fine scale timing signal. This 9 MHz signal drives the Horizontal Counter and Decodes, which produce horizontal blanking and synchronization signals, as well as a two times horizontal signal. This increments the Vertical counter, which is decoded to produce vertically related timing signals. The Blanking and Sync Logic accepts the horizontal and vertical signals and produces Composite Blanking, Composite TV sync, and restart signals to the Reconstructor and the and the Statistical Decoder.

Synchronization of the TV timing to the input data is achieved by means of a Vertical sync pulse from the Rate Buffer. As described earlier, this signal is generated by the recognition of a 30-bit synchronization word in the data. The Vertical Sync pulse sets the Vertical Counter to a state corresponding to the start of vertical blanking.

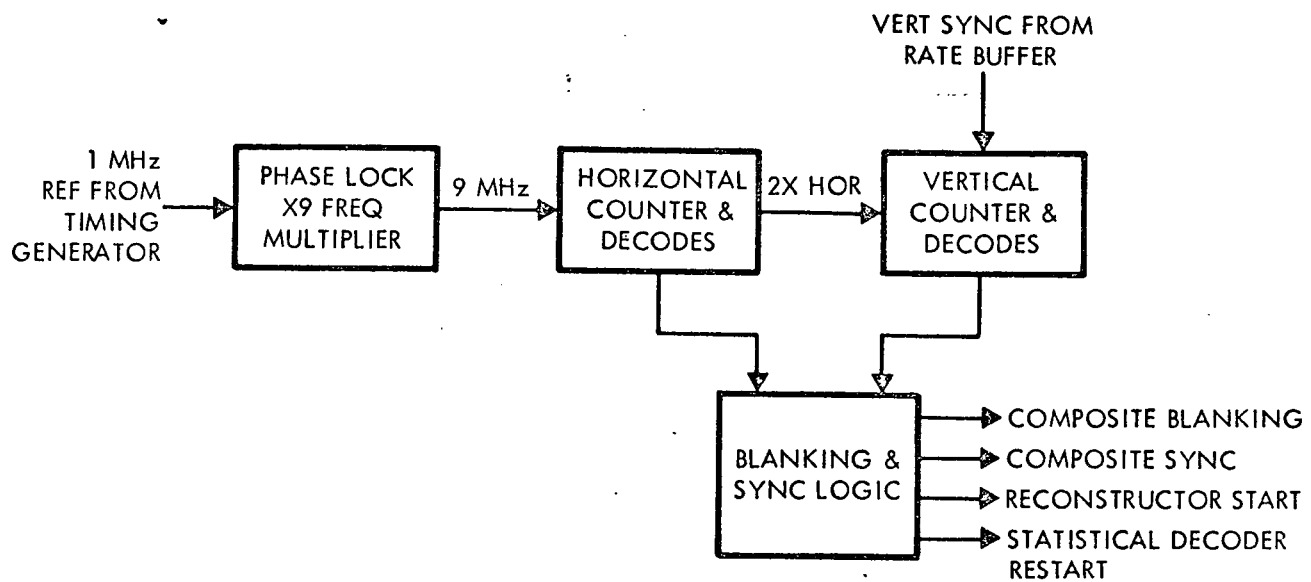
A2-3.10 Digital-to-Analog Converter - A8A1A5 and A8A1A7

This unit converts the six-bit digital data from either the Reconstructor or the Statistical Decoder into analog form. A block diagram is shown in Figure A2-3.10. Selector Gates, controlled by the ZOP (I)/Stat switch route the selected data to the inputs of the D/A Converter unit. The output of the Converter is applied to the Video Amplitude Shaper.

A2-3.11 Video Amplitude Shaper (Expander) - A8A1A6

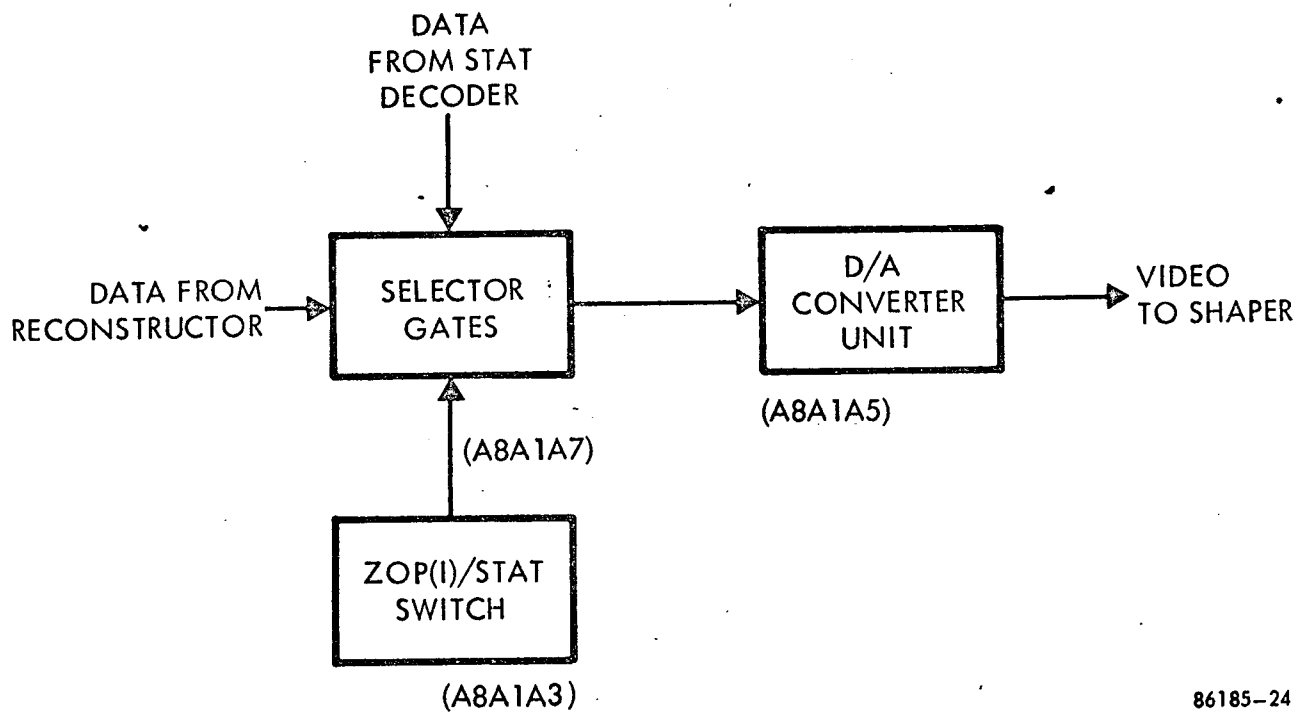
The Receiver Video Amplitude Shaper (Drawing 528742) accepts video information from the D/A Converter, and performs amplitude shaping complementary to that performed in the transmitter. Also, the Shaper has to add the sync signal to the reconstructed video information.

The transfer function of the Receiver Video Amplitude Shaper is piecewise linear, having up to 3 line segments. Each line segment has a greater slope than the previous one so that small amplitude signals are amplified less than large amplitude signals. The transfer function is normally complementary to that of the Transmitter Video Amplitude Shaping so that overall system linearity is preserved.



86185-23

Figure A2-3.9. TV Sync Generator



86185-24

Figure A2-3.10. Digital-to-Analog Converter

The Video Processing Channel consists of amplifiers AR1, AR2, AR3, and AR4 and their associated circuitry. Transistor Q1, amplifier AR5 and reference diode CR5 form the necessary circuit to establish a low output impedance, precision voltage source to which the break points of the linear segments are referenced. Analog switch S1 and reference diode CR5 form the switch circuit which generates the sync signal. The sync signal is added to the video signal by summing resistor R15.

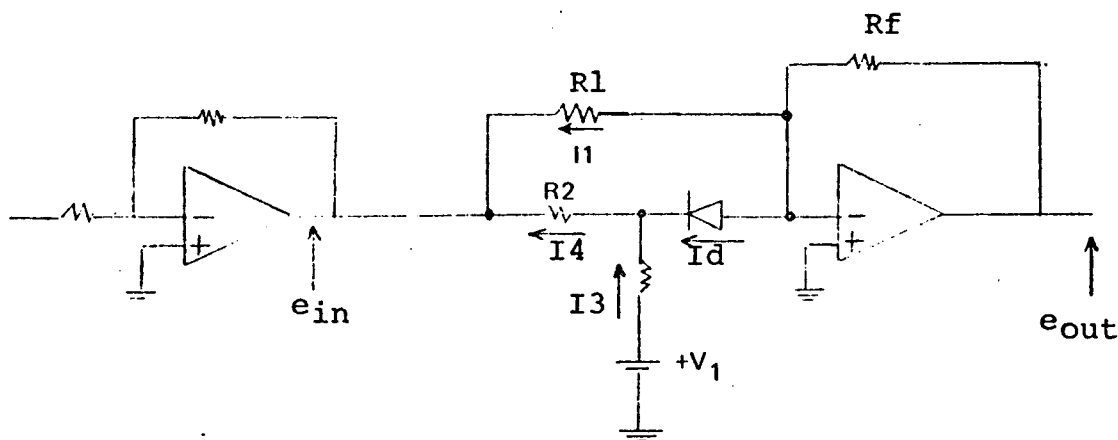
Amplifier AR1 is used as a buffer to isolate the Compressed Video Signal from the Shaping network. Amplifier AR3 is used to sum in the sync signal and as a scaling amplifier to scale the output amplitude of amplifier AR2. Amplifier AR4 is used to drive the output video signal into a 75-ohm system.

Amplifier AR2 and its associated circuitry perform the shaping function on the video. The two break points are determined by the resistive combination of R5, R6, R7, R8, R9, R10, R23, and R24, where R5 and R26 set the bias current for the diodes CR1 and CR2, where R6, R8, and R9 determine the slope of the line segments and where R24, R10, R23, and R9 determine the break points.

From Operational Amplifier fundamentals, the gain of an Operational Amplifier, which determines the slope of a line segment, can be expressed as:

$$G = - \frac{R_f}{R_{in}}$$

From this expression, R_{in} must decrease in value as the video amplitude increase, if the first segment of the piecewise linear transfer function is to have a slope less than the second segment. Consider the following circuit:



If e_{in} is 0 volts, then

$$I_1 = 0$$

$$I_d = 0$$

$$I_3 = I_4$$

$$e_{out} = 0 \text{ volts}$$

For the above conditions, the diode is back biased by V_1 , hence, there is no diode current flowing. As e_{in} goes negative:

$$e_{out} = - \frac{R_f}{R_1}$$

Providing:

$$V_2 \geq 0 \text{ volts}$$

As e_{in} gets more negative, a point will be reached where the drop across R_3 will be equal to or greater than V_1 . At this point the diode will become forward biased and the output voltage will be a function of R_1 , R_2 , and the diode resistance " R_d ." Assume that:

$$I_3 R_3 > V_1$$

$$I_3 \ll I_d$$

then:

$$I_d \approx I_4$$

and

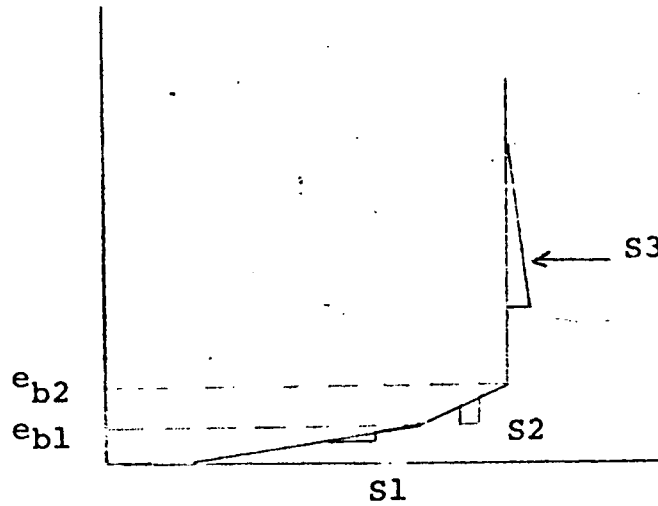
$$e_{out} = \frac{R_f}{R_T}$$

where

$$R_T = \frac{(R_1)(R_2 + R_d)}{R_1 + R_2 + R_d}$$

Therefore, if R_T is less than R_f , then the second segment will have a slope that is greater than the first segment. This is the principle of operation that governs amplifier AR2 and its circuitry as it expands the input video from the D/A Converter.

The 3-segment piecewise linear transfer function is depicted by the following illustration.



Because there are four transfer functions associated with the transmitter, four transfer functions will be required for the receiver. The slopes for the line segments of each are:

1. Linear $s_1 = s_2 = s_3 = 1$

2. $s_1 = \frac{1}{10}$

$s_2 = 1$

$s_3 = 3$

3. $s_1 = 1/4$

$s_2 = 1$

$s_3 = 3$

4. $s_1 = 1/2$

$s_2 = 1-1/4$

Since the performance of the high speed operational amplifiers used is degraded if a gain of 1/10 is used, amplifier AR3 is used to scale the output of AR2 as follows:

$$G_{\text{Total}} = G_1 G_2$$

where

$$G_1 = \frac{R_{f2}}{R_{in2}} \text{ amplifier AR2}$$

$$G_2 = \frac{R_{f3}}{R_{in3}} \text{ amplifier AR3}$$

then

$$G_{\text{Total}} = \frac{R_{f2}}{R_{in2}} \cdot \frac{R_{f3}}{R_{in3}}$$

Thus, for a gain of 1/10, we could have:

$$G = \frac{1}{3.76} \times \frac{1}{2.66} = \frac{1}{10}$$

In this way neither amplifier has to operate with a gain of 1/10. The scaling factor for AR3 is 1/2.66. Therefore, the scaling factor for AR2 for the transfer function line segments are:

$$1. \quad S_1 = 2.66$$

$$2. \quad S_1 = \frac{1}{3.76}$$

$$S_2 = 2.66$$

$$S_3 = 8$$

$$3. \quad S_1 = \frac{1}{1.5}$$

$$S_2 = 2.66$$

$$S_3 = 8$$

$$4. \quad S_1 = 1.33$$

$$S_2 = 3.99$$

The equations relating the circuit components to the slopes and break points are:

$$S_1 = \frac{R_{11}}{R_6}$$

$$S_2 = \frac{R_{11}}{R_{T1}}$$

$$S_3 = \frac{R_{11}}{R_{T2}}$$

where

$$R_{T1} = \frac{(R_6)(R_8 + R_d)}{R_6 + R_8 + R_d}$$

$$R_{T2} = \frac{(R_{T1})(R_9 + R_d)}{R_{T1} + R_9 + R_d}$$

and

$$e_{b1} = V_1 \frac{R_8}{(R_{27} + R_T)}$$

$$e_{b2} = V_1 \frac{R_9}{R_{10} + R_{24}}$$

where

$$V_1 = 4.95 \text{ volts}$$

$$R_T = \frac{(R_{23})(R_{37})}{R_{23} + R_{37}}$$

Since the expander sees the output of the 0 to 5 volt D/A, the break points should be at:

$$1/2 \cdot 5 = 2.5 \text{ volts}$$

and

$$3/5 \cdot 5 = 3 \text{ volts}$$

If one is to observe these break points at amplifier AR2 or AR3, then the break point voltage will have to be scaled appropriately. For instance, if the waveform is observed at the output of AR3 for the transfer function containing line segments of slopes:

$$S_1 = 1/10$$

$$S_2 = 1$$

$$S_3 = 3$$

the first break point would occur at:

$$e_{b1} = \left(\frac{1}{10}\right) 2.5 = .25 \text{ volts.}$$

The second break point would occur at:

$$e_{b2} = .25 + \Delta V (S_2)$$

when $\Delta V = 3 \text{ V} - 2.5 \text{ V} = .5 \text{ volts}$

Then $e_{b2} = .25 + (.5) (1) = .75 \text{ volts}$

It should be understood that the above equations for the break points and slopes are approximations and the actual desired parameters were determined or confirmed empirically.

A2-3.12 Voice Demodulator - A8A1A3

The Voice Demodulator decodes a 20 Kbit serial data stream to reproduce an audio signal. The decoding technique essentially duplicates the delta modulation encoding technique used in the voice encoder. Amplitude expansion is used in the decoder to complement the amplitude compression in the encoder.

The Voice Demodulator (Drawing 520745) contains a high information delta demodulator, which converts a serial bit stream to an analog signal, and a voice expander, which expands the 30 dB range of the incoming signal into the normal 60 dB audio range. The Voice Demodulator reconstructs the original voice inputted to the Voice Encoder.

The high information delta demodulator operates similarly to the high information delta modulator in the Voice Encoder. The Data Input is sampled and held by a D-type flip-flop whose output is to a two stage shift register. Decoding circuitry on the shift register outputs generates commands to a 4-bit Shift Left - Shift Right (SLSR) register in accordance with the table as follows:

(A follows B follows C)

Shift Register State	SLSR Command
$A \neq B \neq C$	Shift Left
$A \neq B = C$	Shift Left
$A = B \neq C$	Hold
$A = B = C$	Shift Right

As in the Voice Encoder, the SLSR shifts ones in from the left and zeroes in from the right. Thus, a shift right command will double the binary weight of the SLSR output and a shift left will half it. The data stream is thus translated to this four bit (plus a sign bit) binary code identically to the Voice Encoder. This output is then applied with proper weighting to the input of an integrator to reproduce the analog signal. The integrator has a dc feedback path to prevent latchup due to bias and offset currents.

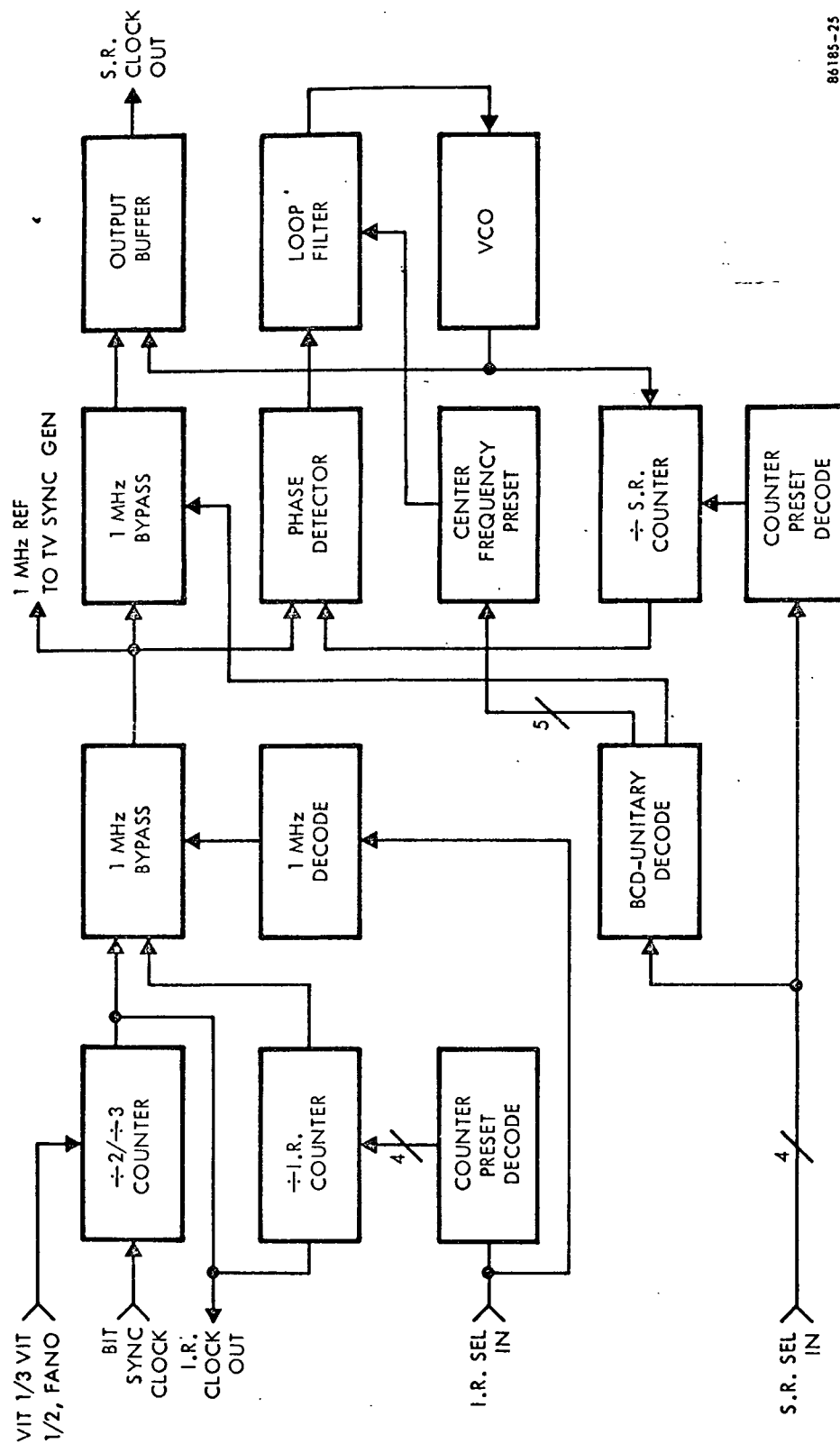
The integrator output goes to the input of the voice expander (or directly out, switch selectable). A nonlinear input resistance at the input to an op amp produces a transfer function which is the inverse of the voice compressor transfer function, described in the Voice Encoder detail operation. The original voice input signal is reproduced at the output of this amplifier. A switch at the op amp output serves to switch the driving point impedance to provide equal amplitude signals into either 50 ohms or 600 ohms.

A2-3.13 Timing Generator - A8A1A4

The receiver timing circuit shown in Figure A2-3.13 generates an information rate clock synchronized with the Bit Synchronizer clock and a 1 to 9 MHz sample rate clock output, which is selected by the front panel Sample Rate selector switch.

The receiver timing circuits accept the bit synchronizer output clock and the rate line (i.e., Viterbi 1/3, Viterbi 1/2, or Fano) from the front panel switch. The bit sync clock is divided by 3 if the code switch is set to 1/3, and it is divided by 2 if the code switch is set to Viterbi 1/2 or Fano positions. The divide is accomplished by a programmable divide by two/divide by three counter. The output of this counter is the information rate, and it is buffered and sent out to the system. If the Group Sync input is selected from an external source, the external clock becomes the information rate clock.

A one-megahertz signal is generated by dividing the information rate clock by the information rate, and is used as a reference frequency to operate to phase-lock loop which generates the sample rate clock. A 4-bit adder has one set of inputs preset to a count of 10. The other four inputs are operated with the BCD inverse of the information rate, which comes from the front panel information rate switch. Thus, the output of this 4-bit adder is a number which is



86185-25

Figure A2-3.13. Receiver Timing Block Diagram

equal to 10 minus the 10's complement of the information rate setting of the front panel switch. The 4-bit synchronous up counter is operated by the information rate from the information rate down counter circuit, and when this decade counter reaches a count of 9, its carry output is routed back to the load input, and the number generated by the 4-bit adder is loaded into the data inputs of the counter. The counter thus acts as a divide by N counter, where N is the information rate. Since it is clocked by the information rate, the output will thus be one-megahertz. When the information rate switch is set to a one megahertz, this position is decoded and in this mode the synchronous counter is bypassed so that the one-megahertz information rate is routed directly to the reference input of the sample rate phase-lock loop. The one megahertz reference is also routed to the TV Sync Generator for a timing reference.

The operation of the sample rate phase-lock loop in the receiver is identical to the operation of the sample rate loop in the transmitter. For details on its operation, refer to Paragraph A2-2.11.

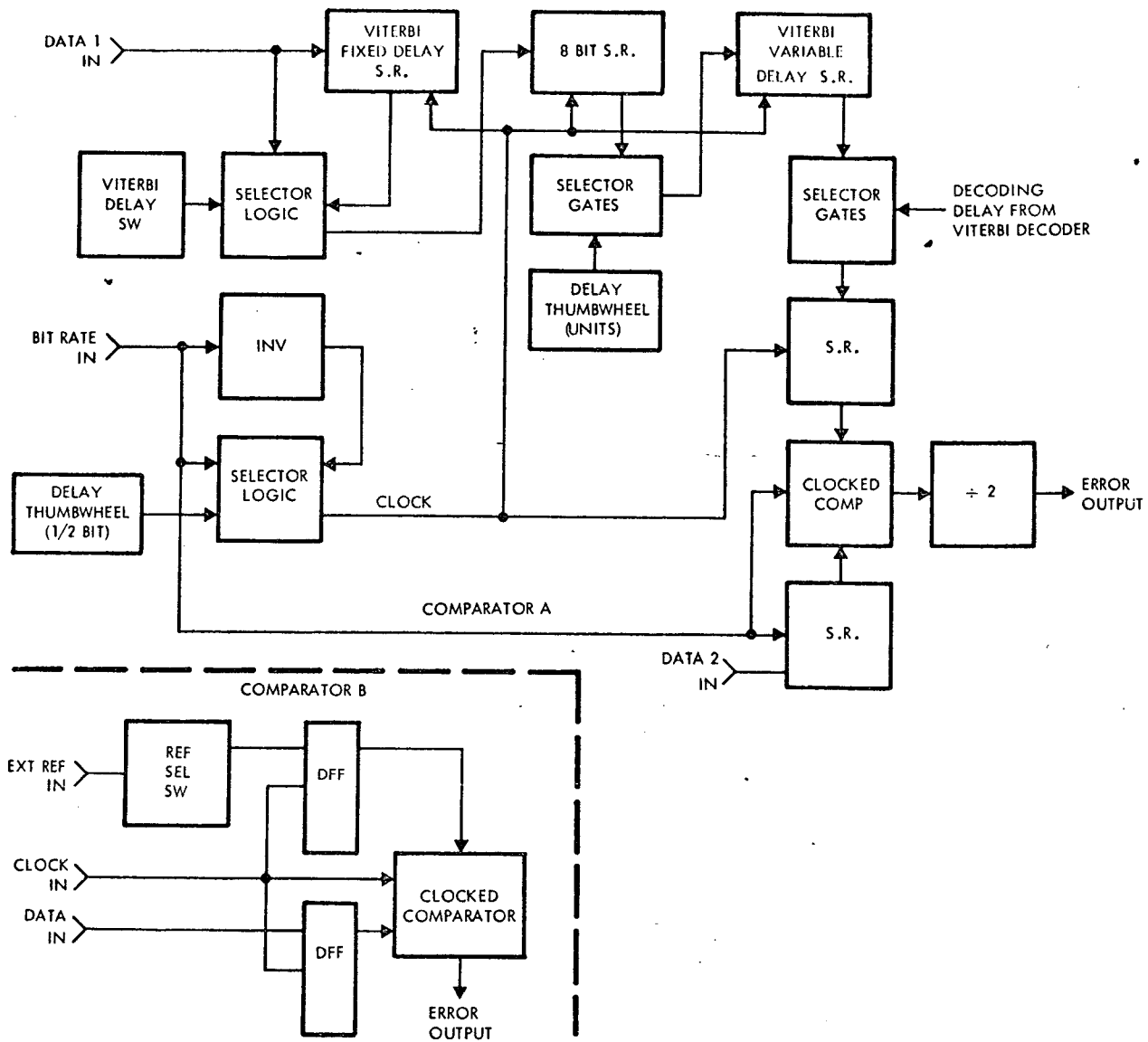
A2-3.14 Bit Comparators - A8A9A1

In order to perform tests and determine the bit error rate at the decoder inputs and outputs, a pair of bit comparators are provided. Comparator A will be used to measure bit error rates at the decoder inputs and at the Viterbi output. Comparator B will provide error measurements at the Fano output.

A block diagram of Comparator A is shown in Figure A2-3.14. This comparator is used to measure bit error rates at the input to the decoder or at the Viterbi decoder output. The former is accomplished by delaying the data at the output of the coder by an amount equal to the delay that results from the transmission link and the bit synchronizer. The data can be delayed by $1/2$ bit increments up to a maximum of $7-1/2$ bits. This is accomplished by selecting either BIT RATE or $\overline{\text{BIT RATE}}$ to strobe an 8-bit shift register. For either 0 or $1/2$ -bit delay the output of the first stage of the shift register is selected. For either 1 or $1-1/2$ -bits delay, the output of the second stage of the shift register is selected and so forth up to $7-1/2$ -bits delay.

At this point the data from the transmission link and the delayed data must line up within $1/2$ bit. To assure further alignment they are each strobed through shift registers by the complement of the clock that was used to strobe the shift register. This ensures that the two data inputs to the comparator are aligned in time so that they can be compared accurately. The errors out of the comparator are divided by two in a counter to provide pulses which can be reliably counted.

In order to measure the bit error rate at the Viterbi output, the data at the input to the coder is delayed by an amount equal to the combined delays of the coder, bit synchronizer and the Viterbi decoder. This is accomplished by incorporating a fixed delay prior to the previously mentioned 8-bit shift register, and a variable delay (0-28 bits in 4-bit increments) following the same register. The fixed delay along with the 8-bit shift register (controlled from the front panel) compensates for the coder and bit synchronizer delays and the Viterbi fixed delay. The following variable delay is controlled by lines from the Viterbi to match the Viterbi variable delay with the decoding delay.



86185-26

Figure A2-3.14. BIT Comparators

The bit comparator is placed in the Viterbi mode or decoder input mode by the "Viterbi Decoder Delay-In/Out" switch (S7) on the front panel.

Comparator B also shown in Figure A2-3.14 is used for measuring the error rate at the output of the Fano decoder. Since the Fano introduces a delay of several thousand bits, it is not practical to delay the coder input signal to form the reference. The simplify implementation and to allow error rate measurements while transmitting TV data, the PCM channel is used for these error rate measurements. At the transmitter (multiplexer) all 1's or all 0's are inserted in place of the PCM input by means of a switch. At the receiver, a similar setting of a switch is required ("1" or "0"). The information, either all 1's or all 0's compared to the receiver switch to generate the error count. An external reference can also be supplied. Note that errors are not divided by two in this comparator, due to the lower design speed.